OPEN COIL BOX ANNEALED STEEL COOLING ANALYSIS

by

Louis Lawrence Gambogi III

BSME, University of Pittsburgh, 2001

Submitted to the Graduate Faculty of

School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science Mechanical Engineering

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Louis Lawrence Gambogi III

It was defended on

July 16, 2003

and approved by

Dr. Richard S. Dougall, Emeritus Professor, Dept. of Mechanical Engineering

Dr. Dipo Onipede, Assistant Professor, Dept. of Mechanical Engineering

Thesis Advisor: Dr. Patrick S. Smolinski, Associate Professor, Dept. of Mechanical Engineering

ABSTRACT

OPEN COIL BOX ANNEALED STEEL COOLING ANALYSIS

Louis Lawrence Gambogi III, M.S.

University of Pittsburgh, 2003

Steel sheet is used for many products. Two common uses are in the production of household appliances and automotive body panels. Typically, the steel is formed to a desired thickness and then external coatings are applied. First a hot then a cold rolling process forms the steel sheet used for these applications. For handling convenience, the sheet metal is wrapped in a cylindrical coil. The term open coil refers to steel sheet that is wound into a coil with intertwined wire separating each successive wrap. The coil separation allows gas to circulate between the sheets during annealing. A gas composed of 8% hydrogen and 92% nitrogen is used to reduce the carbon content of the steel, which allows better coating adhesion. After annealing, the coil is cooled in the box furnace and then set on a cooling bed where cooling of the coil is done by forced convection and experimental data was obtained. Temperature and air flow were recorded. A computer model was developed to calculate the temperature distribution in the open coil during convection cooling. A finite difference model, based on the energy balance between air and steel gave the governing equations for the computer program. The program was compared to

the experimental data and used to calculate the temperature distribution in a coil over time. The results were used to approximate the required cooling time of a coil.

DESCRIPTORS

Coil Annealing	Convection
Finite Difference	Heat Transfer

ACKNOWLEDGEMENTS

I would like to thank US Steel Corporation for their support of this project. I would like to thank Dr. Edward Patula, Dr. Joseph DeFilippi, Dr. Joseph Gallenstein and Dr. Eugene Nikitenko of the US Steel Research and Technology Center for giving me the opportunity to work on this project. Appreciation is also given to Achilles Vassilicos, Dave Stewart and Patrick Wheeler of the US Steel Research and Technology Center, as well as the personnel at the US Steel Irvin Works. I especially would like to thank my advisors, Dr. Patrick S. Smolinski, Dr. Dipo Onipede Jr. and Dr. Richard S. Dougall for their guidance during the course of this project.

I dedicate this project to my family and loved ones. Their love, guidance and support have given me the drive to follow new paths and seek new challenges. To make them proud is my greatest desire.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The process of open coil annealing (OCA) is used to prepare steel sheet for further processing. The OCA process anneals and de-carbonizes the steel in preparation for shaping and finishing by original equipment manufacturers. The steel sheet that undergoes this process is used primarily in the appliance and automotive industries. Because of its use, the steel sheet must be highly formable and have good surface chemical properties.

In OCA, a coil of steel is wound with two intertwined 0.035 inch wires between each sheet. The wires are used to space the wraps of the coil, thus creating an "open coil". This spacing allows air flow along all of the surfaces of the steel coil. Annealing is done to soften the steel by relaxing the strain hardening caused by cold rolling. The de-carbonization is done simultaneously with the annealing. Lower carbon content allows better adhesion of surface coatings. De-carbonization is accomplished by circulating a specialized gas through the open wraps. After annealing and de-carbonization, gas flow continues to circulate within the furnace to initiate cooling. The coil is then removed from the furnace and placed on a cooling bed. Cooling at this point is done by forced convection over a large floor fan. Conversely, if the coil is removed from the cooling bed too soon, it will be too hot to place on the rubber padded coil winder base.

A fan draws air down from the top of the coil, axially through the bottom of the coil down through the cooling bed grid base. All coils are placed on the cooling bed and there is some temperature variation throughout the coil. The top is usually much cooler than the bottom

because the coil is placed on the floor for a period of time before being put on the cooling bed. The cooling time and final temperature of the coil when removed from the cooling bed vary with coil dimensions and initial temperatures.

Measurements have been taken to evaluate air velocity and coil temperature at the cooling bed base, however, there has not been any extensive testing or modeling to determine the temperature distribution in the coil. Development of a model would result in increased productivity by providing information on the minimum cooling time. This model would take into consideration thermal parameters as well as coil geometry. The model would be used to investigate the temperature distribution in the coil so that a better understanding of the cooling behavior can be attained.

2.0 LITERATURE SURVEY

Before a numerical model of the heat transfer in the coil was created, a literature search was done to study the previous work on the subject of steel coil cooling. The literature search led to two articles that provided some information that inspired ideas for modeling.

2.1 Finite Element Analysis of Hot Rolled Coil Cooling

A unit layer model created by Park, Hong and Baik [1] for the equivalent thermal conductivity of layered steel strips examines the equivalent thermal conductivity as a function of strip thickness, surface characteristics and compressive stresses. Finite element analyses of the cooling of a hot rolled coil were carried out under various cooling conditions investigating the thermal conductivity in the radial direction. A new calculation procedure was developed using the ABAQUS software. The radial compressive thermal stress was taken into account in the calculation of the equivalent thermal conductivity in the radial direction in the coil.

2.2 A Two-Equation Analysis of Convection Heat Transfer in Porous Media

This study examined the flow through a porous medium, a packed bed, between the parallel walls. Darcy flow was imposed and fully developed heat transfer was assumed. A two-

equation analysis on the convection heat transfer in porous media was developed by Zhang and Huang [2]. One of the parallel walls was heated uniformly, while the other was exposed to the ambient air. The porous system was characterized by circular unit cells. Conduction and convection were assumed in the general equations.

3.0 GOVERNING EQUATIONS

Methods developed by Patankar [3] were used to examine the energy balance between a fluid and solid. By investigating a control volume, the energy balance equations between the materials are developed. These energy balance equations are derived from partial differential equations. The discretization methods evolved from the partial differential equations. The iterative solution, as well as the tri-diagonal matrix algorithm, was developed from Patankar's work.

The energy balance equations between a moving fluid and a solid were developed by Patankar [4] and used as the basis for a numerical model to examine the cooling of a steel coil. The problem statement for a fluid and wall is shown in Figure 1. The fluid flow has a velocity, u, parallel to the walls of a tube while heat, q, is released to the fluid.

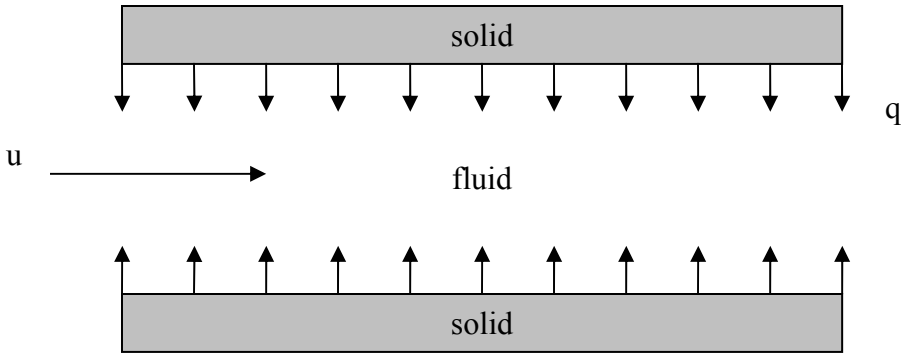


Figure 1. The boundary condition between a fluid and wall interface.

The cooling model was developed based on the theory derived for temperature distribution in a channel and wall under transient conditions with heat transfer between the wall and fluid. In the model one-dimensional flow with negligible axial conduction was assumed. The following partial differential equations are used to describe the energy balance for the fluid and wall, respectively,

$$\left(\rho_f c_f \frac{\partial T}{\partial t}\right)+\left(\rho_f c_f u \frac{\partial T}{\partial x}\right)=h\frac{P_h}{A}\left(T_w-T\right) \qquad (1)$$

$$\rho_w c_w \frac{\partial T_w}{\partial t}=h\frac{P_h}{A}\left(T-T_w\right) \qquad (2)$$

where,

ρ_f - Density of the fluid.

c_f - Specific heat of the fluid.

T - Temperature of the fluid.

t - Time.

u - Velocity of the fluid.

x - Axial coordinate.

h - Convective heat transfer coefficient.

P_h - Hydraulic diameter.

A - Surface area perpendicular to the fluid flow.

T_w - Temperature of the wall.

T - Temperature of the fluid.

ρ_w - Density of the wall.

c_w - Specific heat of the wall.

T_w - Temperature of the wall.

For the fluid and wall, all parameters are assumed constant. The hydraulic perimeter is the perimeter of the surface perpendicular to the fluid flow and convective heat transfer coefficient must be determined experimentally. The governing equations were the basis for the development of the numerical model.

4.0 NUMERICAL METHODS

4.1 Finite Difference Approximation

The energy balance equations are applied to a finite difference model. From these equations, discretized equations for both air and steel are developed. Figure 2 shows the finite difference grid for a discrete model of the coil. In the finite difference model, each successive air channel and steel wrap, has an equation representing the energy balance. Starting from the innermost steel wrap, which is along the inner radius of the coil, i = 1, the channels and wraps are numbered sequentially to the outer wrap of the coil. From this convention all odd numbers of i, represent steel wraps, while all even numbers represent air channels. Each air channel and steel wrap is divided into sections or nodes, j, along the axial direction of the coil. Air flow is along the axial direction of the coil.

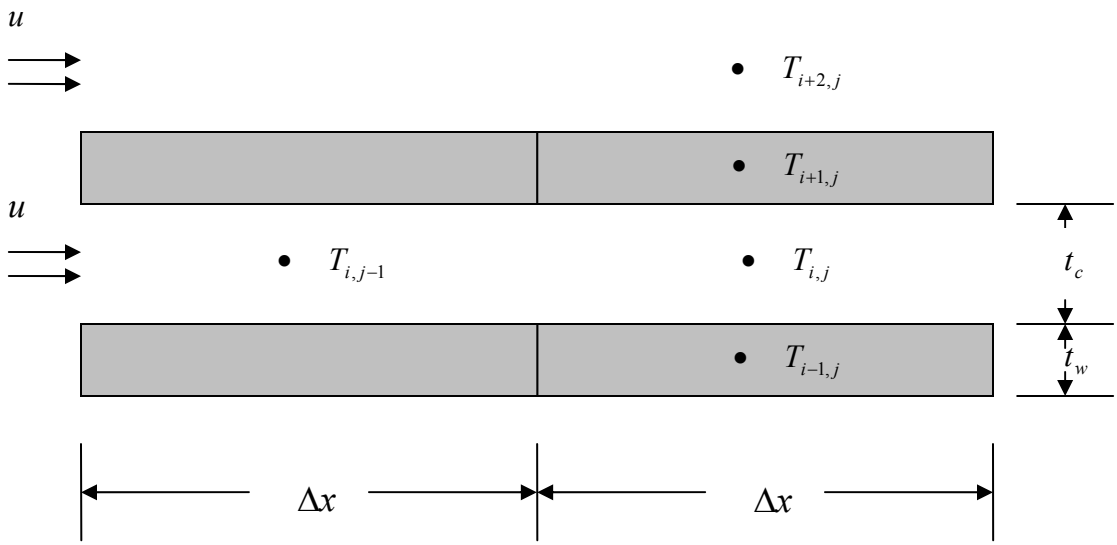


Figure 2. Finite Difference Model for an air channel and a steel wrap in a coil.

Using the notation given in Figure 2, a difference equation, form of equation *(1)*, for air node, i, j is given by,

$$\left(\rho_a c_a A_i \frac{\partial T_{i,j}}{\partial t}\right)+\left(\rho_a c_a A_i u \frac{\partial T_{i,j}}{\partial x}\right)= hP_{out,i}\left(T_{i+1,j}-T_{i,j}\right)+ hP_{in,i}\left(T_{i-1,j}-T_{i,j}\right) \qquad (3)$$

where, $P_{out,i}$ is the outer perimeter of the air channel, i, and $P_{in,i}$ is the inner perimeter. Substituting a backward difference operator for the time derivative term and an upwind difference operator for the spatial derivative gives,

$$\left[\left(\rho_a c_a A_i \frac{\Delta x}{\Delta t}\right)+\left(\rho_a c_a A_i u\right)+\left(hP_{out,i}\Delta x\right)+\left(hP_{in,i}\Delta x\right)\right]T_{i,j} =$$

$$\left(hP_{out,i}\Delta x\right)T_{i+1,j}+\left(hP_{in,i}\Delta x\right)T_{i-1,j}+\left(\rho_a c_a A_i \frac{\Delta x}{\Delta t}\right)T_{i,j}^{O}+\left(\rho_a c_a A_i u\right)T_{i,j-1} \qquad (4)$$

where the superscript O is used to denote temperatures at the previous time step. The energy balance for the inner steel wrap is created in a similar manner. Substituting the difference operators into equation *(2)*, the difference equation for node $i+1, j$ is given by,

$$\left(\rho_s c_s A_{i+1} \frac{\Delta x}{\Delta t}\right)\left(T_{i+1,j}-T_{i+1,j}^{O}\right)=hP_{out,i+1}\Delta x\left(T_{i+2,j}-T_{i+1,j}\right)+hP_{in,i+1}\Delta x\left(T_{i,j}-T_{i+1,j}\right) \qquad (5)$$

Collecting terms and rearranging equation *(5)* gives the discrete equation,

$$\left[\left(\rho_s c_s A_{i+1} \frac{\Delta x}{\Delta t}\right)+\left(hP_{out,i+1}\Delta x\right)+\left(hP_{in,i+1}\Delta x\right)\right]T_{i+1,j} =$$

$$\left(hP_{out,i+1}\Delta x\right)T_{i+2,j}+\left(hP_{in,i+1}\Delta x\right)T_{i,j}+\left(\rho_s c_s A_{i+1} \frac{\Delta x}{\Delta t}\right)T_{i+1,j}^{O} \qquad (6)$$

Equations *(4)* and *(6)* are written for each node of the finite difference model of the coil and are solved by the computer program to obtain the temperature distribution in the coil.

4.2 Tri-Diagonal Matrix Algorithm

Equations *(4)* and *(6)* can be simplified to form a set of tri-diagonal linear, algebraic equations of the type [5],

$$a(i)T_i = b(i)T_{i+1} + c(i)T_{i-1} + d(i) \qquad (7)$$

For a number of air channels and steel wraps, $i = 1, n$ at a given axial position, applying equation *(7)* would form a tri-diagonal matrix of the form in Figure 4.

$$\begin{bmatrix} a(1) & b(1) & 0 & 0 & 0 & 0 \\ c(2) & a(2) & b(2) & 0 & 0 & 0 \\ 0 & c(3) & a(3) & b(3) & 0 & 0 \\ 0 & 0 & ... & ... & ... & 0 \\ 0 & 0 & 0 & ... & ... & ... \\ 0 & 0 & 0 & 0 & c(n) & a(n) \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ ... \\ ... \\ T_n \end{Bmatrix} = \begin{bmatrix} d(1) \\ d(2) \\ d(3) \\ ... \\ ... \\ d(n) \end{bmatrix}$$

Figure 3. Tri-diagonal matrix.

For an air channel, applying the form of equation *(7)* to equation *(4)* gives,

$$a(i) = \left[\left(\rho_a c_a A_i \frac{\Delta x}{\Delta t} \right) + \left(\rho_a c_a A_i u \right) + \left(hP_{out,i} \Delta x \right) + \left(hP_{in,i} \Delta x \right) \right] \quad (8)$$

$$b(i) = \left(hP_{out,i} \Delta x \right) \quad (9)$$

$$c(i) = \left(hP_{in,i} \Delta x \right) \quad (10)$$

$$d(i) = \left(\rho_a c_a A_i \frac{\Delta x}{\Delta t} \right) T_{i,n}^{O} + \left(\rho_a c_a A_i u \right) T_{i,n-1} \quad (11)$$

For a steel wrap, applying the form of equation *(7)* to equation *(6)* gives,

$$a(i) = \left[\left(\rho_s c_s A_i \frac{\Delta x}{\Delta t} \right) + \left(hP_{out,i} \Delta x \right) + \left(hP_{in,i} \Delta x \right) \right] \quad (12)$$

$$b(i) = \left(hP_{out,i} \Delta x \right) \quad (13)$$

$$c(i) = \left(hP_{in,i} \Delta x \right) \quad (14)$$

$$d(i) = \left(\rho_s c_s A_i \frac{\Delta x}{\Delta t} \right) T_{i,n}^{O} \quad (15)$$

In the case of a number of equations, n, for $i = 1$, equations *(14)* and *(15)* become,

$$c(i) = 0 \quad (16)$$

$$d(i) = \left(\rho_s c_s A_i \frac{\Delta x}{\Delta t} \right) T_{i,n}^{O} + \left(hP_{in,i} \Delta x \right) T_{room} \quad (17)$$

where T_{room} is the ambient temperature. For $i = n$, equations *(14)* and *(15)* become,

$$b(i) = 0 \tag{18}$$

$$d(i) = \left(\rho_s c_s A_i \frac{\Delta x}{\Delta t} \right) T_{i,n}^{O} + \left(hP_{out,i} \Delta x \right) T_{room} \tag{19}$$

For the numerical model, constants a, b and c are all independent of time, while d changes with time. All constants are dependent on the coil geometry and thermal properties.

4.3 Computer Model

The energy balance equations, in the form of a set of tri-diagonal linear, algebraic equations, were used as the basis for a numerical model. This model used input data to set up preliminary calculations, such as dimensions, initial temperatures and various arrays. The time step loop included an embedded axial node loop, for nodes along the axial width of the coil. Each axial node loop executed the calculations for each set of axial nodes. The time step iterates over the number of time steps in the analysis. Each set of nodal temperatures for a time step were used as the previous temperatures for the next time step. A flow chart of the functions used for the model is given in Figure 4.

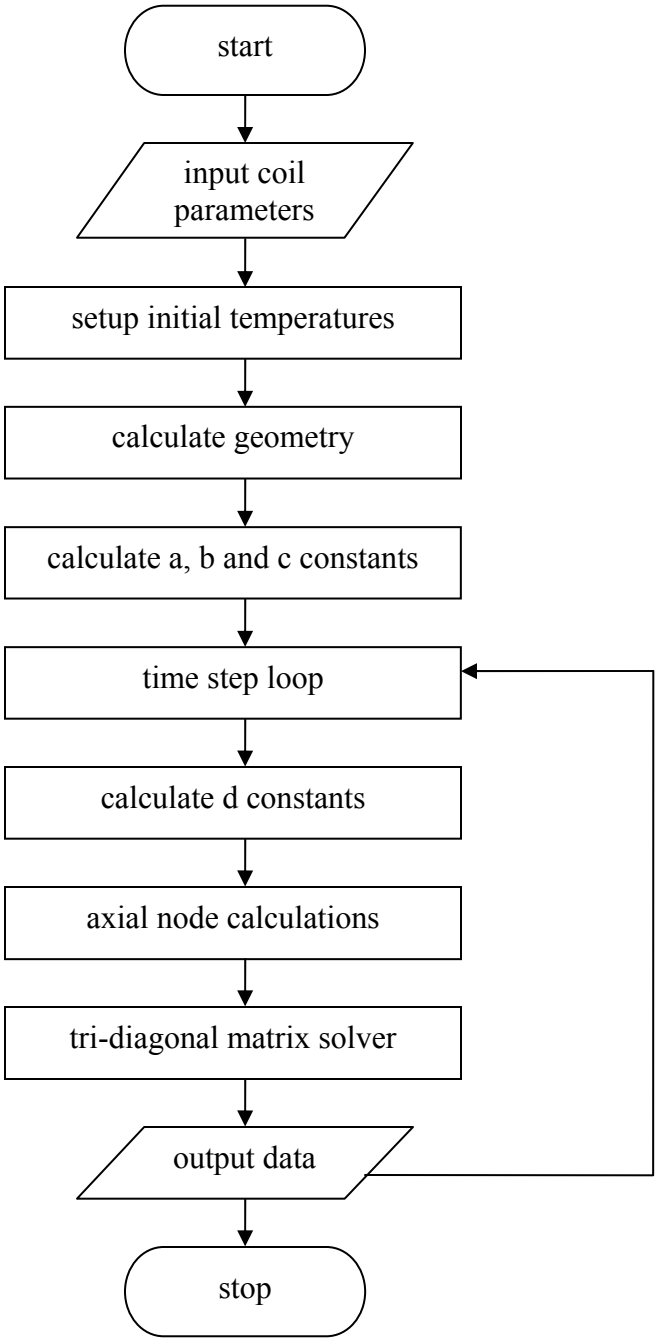


Figure 4. Flowchart for numerical computer model.

Details of the programming code including the individual files are included in appendix H. They are listed in order with Figure 4.

5.0 FIELD TESTING

Data was recorded on steel coils to be used to validate the results from the computer model. The data plotted the temperature at different locations in a coil over time while it was being cooled by forced convection. Air pressure readings were taken in an attempt to measure the air flow through the interlayer spacing of the coil. Coils were usually placed on the floor for a period of time before being placed on the cooling bed. This resulted in a non-uniform initial temperature distribution in the coil when it was placed on the cooling bed.

5.1 Data Acquisition

Thermocouples were positioned radially from the inner to outer radius of a coil on both the top and bottom as it laid on the cooling bed. Six evenly spaced thermocouples were placed from the inner to outer radius under the bottom of the cooling bed. The inner or first thermocouple is positioned near the outer edge of the 34 in. (863.6 mm) diameter cover plate which is at the center of the cooling bed. The thermocouples were referred to by radial position. The first was position 1 which was about 1 in. (25.4 mm) from the inner radius of the coil. Positions 2 3, 4, 5 and 6 were 9, 15, 21, 27 and 33 in. (228.6, 381.0, 533.4, 685.8 and 838.2 mm) from the outside edge of the inner plate, respectively. The thermocouple at position 6 was close to the outside radius of the cooling bed. Six more thermocouples were placed on top of the coil

directly above the six on the bottom. Depending on the coil radius, three or four of the six thermocouples were used for a given coil across both its top and bottom edges. The last one used was the position nearest the outside radius of the coil. One or two in between were used for mid-radius positions.

On the bottom of the coil, a set of static and dynamic pressure pitot tubes were positioned next to each thermocouple position. Three of the six sets of pitot tubes were used to read the pressures near the inner, middle and outer radius. An inclined manometer was used to measure the difference between the dynamic and static column pressures. These pressure readings were converted to air velocity readings using Bernoulli's equation.

5.2 Field Temperature Data

Field data was taken of the temperature histories for coils with a varying range of dimensions. Comparison of the temperature history was made to determine the effect coil parameters had on cooling times and patterns.

Table 1 gives the data for seven coils whose temperatures were recorded. "Gage" refers to the thickness of individual steel wraps. "Width" is the distance from the top to bottom of the coil as it lies on its side on the cooling bed. "O.D. Open" refers to the outer diameter of the open coil. "Record Time" is the total time that data was being recorded for a coil. "Sample Rate" is the time between each individual recording of a temperature by the data acquisition equipment. "Room Temp." is the ambient room temperature. "Top Temp." is the initial temperature of a point in the middle on top of the coil and "Bottom Temp." refers to the initial temp at a point in the middle on the bottom of the coil.

Table 1. Coil Field Data.

Case	Gauge (in./mm)	Width (in./m)	O.D. Open (in./m)	Record Time (s)	Sample rate (s)	Room Temp. (^{O}F/^{O}C)	Top Temp. (^{O}F/^{O}C)	Bottom Temp. (^{O}F/^{O}C)
1	0.0201/ 0.510	48.50/ 1.245	101.5/ 2.578	2988	6	89.8/ 32.11	98/ 36.67	230/ 110.0
2	0.0332/ 0.843	42.68/ 1.084	98/ 2.489	2988	12	87.8/ 31.00	150/ 65.56	190/ 87.78
3	0.0240/ 0.609	60.56/ 1.538	N/A	2988	12	87.6/ 30.89	90/ 32.22	210/ 98.89
4	0.0377/ 0.957	34.75/ 0.882	89/ 2.260	1434	12	90.7/ 32.61	120/ 48.89	150/ 65.56
5	0.0240/ 0.609	60.56/ 1.538	95/ 2.413	846	12	86.4/ 30.22	95/ 35.00	190/ 87.78
6	0.0402/ 0.102	47.19/ 1.199	112/ 2.845	2946	12	88.8/ 31.56	145/ 62.78	205/ 96.11
7	0.0332/ 0.843	55.60/ 1.412	N/A	2988	6	84.6/ 29.22	85/ 29.44	195/ 90.56

5.3 Temperature Histories

Temperatures at several points on the top and bottom of a coil were recorded using a data recorder with a built-in signal conditioner. The data was recorded and plotted using a spreadsheet. Figure 5 gives an example of the temperature history at position 1 of Case 1, a medium width coil.

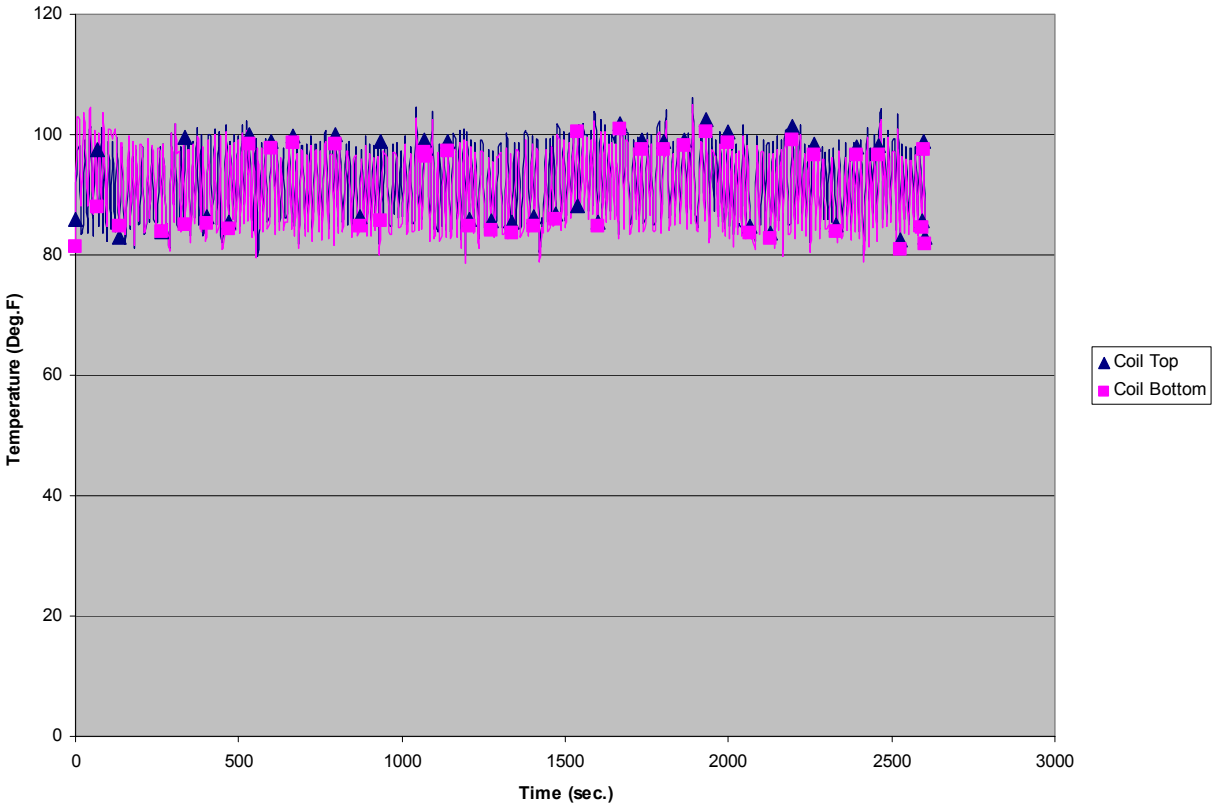


Figure 5. Temperature history at position 1 for Case 1.

In Figure 5, the temperatures for both the coil bottom and top are near the ambient temperature during the entire time the coil was cooled. At this location both the top and bottom of the coil were close to the inner wrap of the coil, were heat was released to the ambient air by free convection. By the time the coil was placed on the cooling bed and temperature recording began, this position may have had time to cool to the ambient temperature.

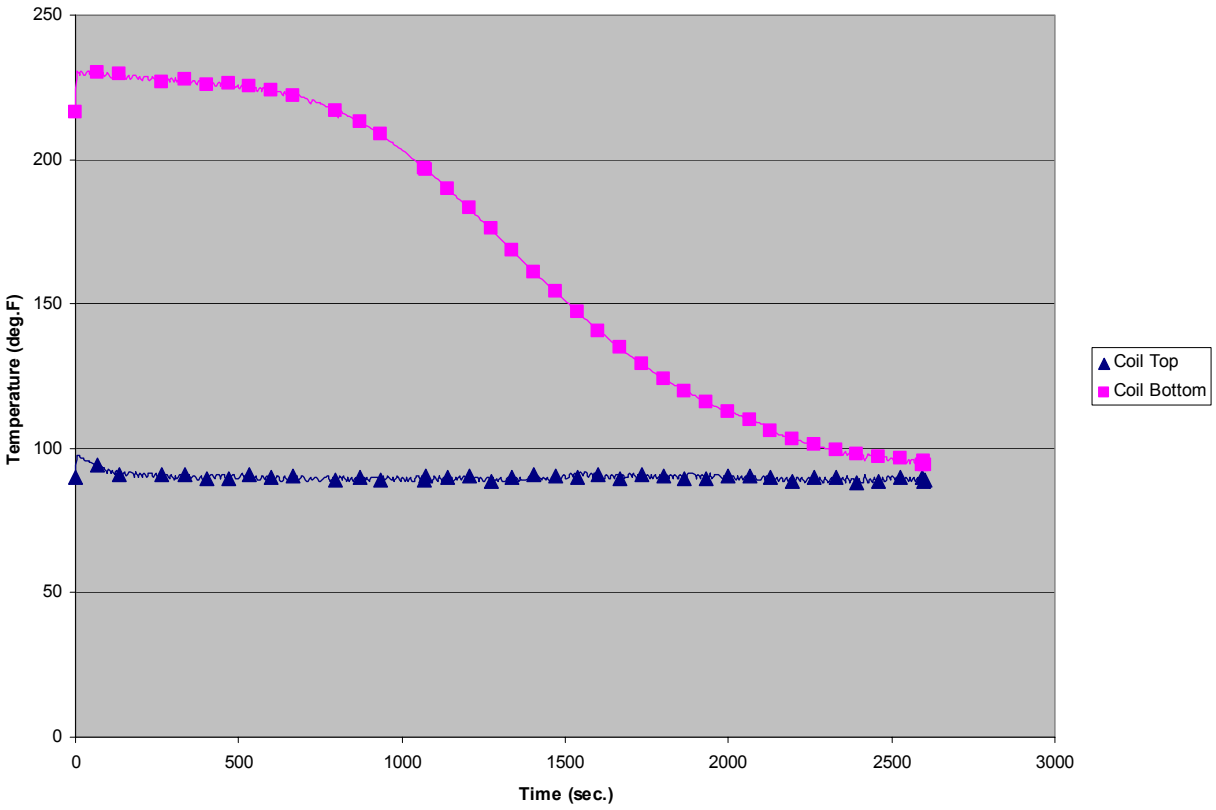


Figure 6. Temperature history at position 5 for Case 1.

Figure 6 gives the temperature history at position 5 for Case 1. At this location, there was a large difference in the initial top and bottom coil temperatures. The initial top temperature was

approximately 210 ^{O}F while the bottom was about 90 ^{O}F. This type of temperature history was common for many coils at the mid-radius position. The temperature history at position 3 shows a curve very similar to that of Figure 6. The temperature histories at positions 1, 3 and 5 for Case 1 are given in Appendix A. These three positions were chosen to measure temperatures at the inner, middle and close to the outer radius of the coil.

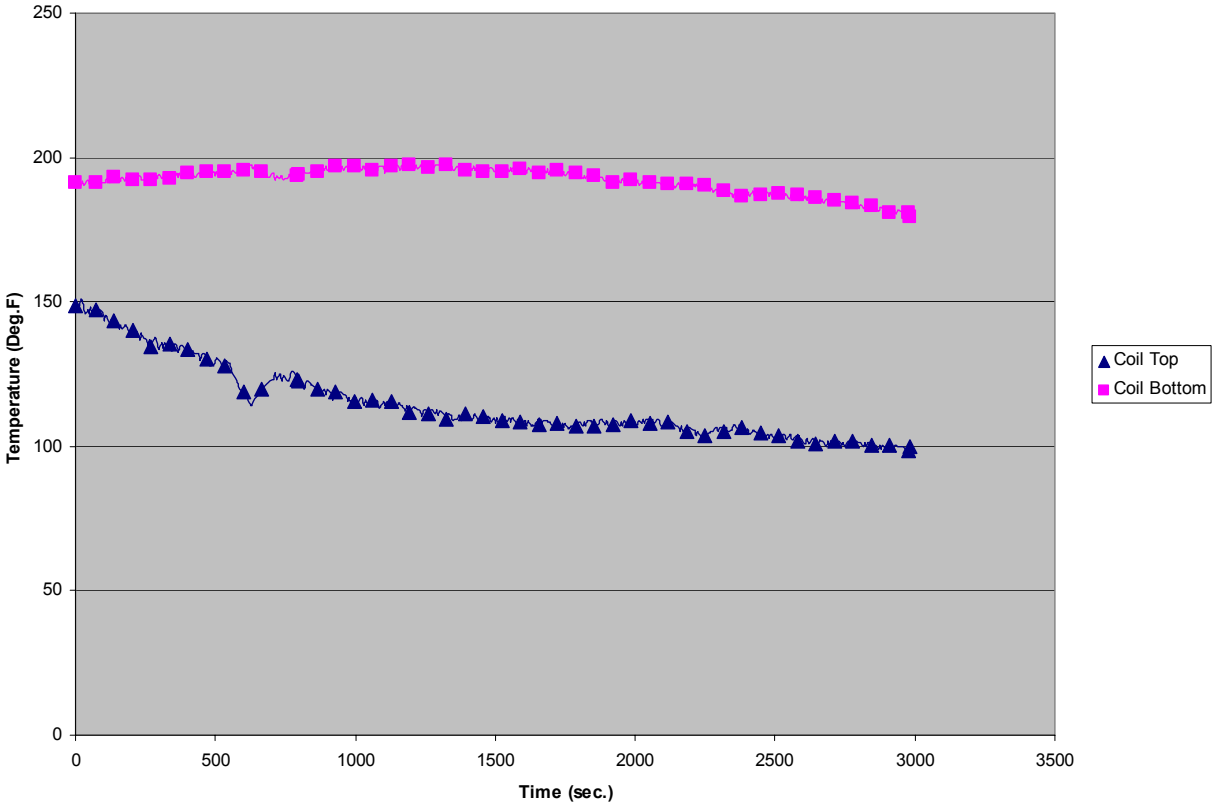


Figure 7. Temperature history at position 4 for Case 2.

Figure 7 gives the temperature history at position 4 for Case 2. This is another medium width coil. In Figure 7, the bottom and top initial temperature difference is smaller than

that of case 1 in Figure 6. From this, it can be surmised that the coil in Case 2 sat on the floor for a shorter period of time then that of Case 1. Temperature histories at positions 2, 3, 4 and 5 for Case 2 are given in Appendix B. Position 1 was not used for Case 2 because the coil was placed off center from the center plate on the cooling bed. Because of this, the thermocouple at this position did not make contact with the coil. The thermocouples at positions 2, 3, 4 and 5 in Case 2 measured temperatures along mid-radii of the coil. Since the coil was off center, the exact radial distance for each position could not be determined.

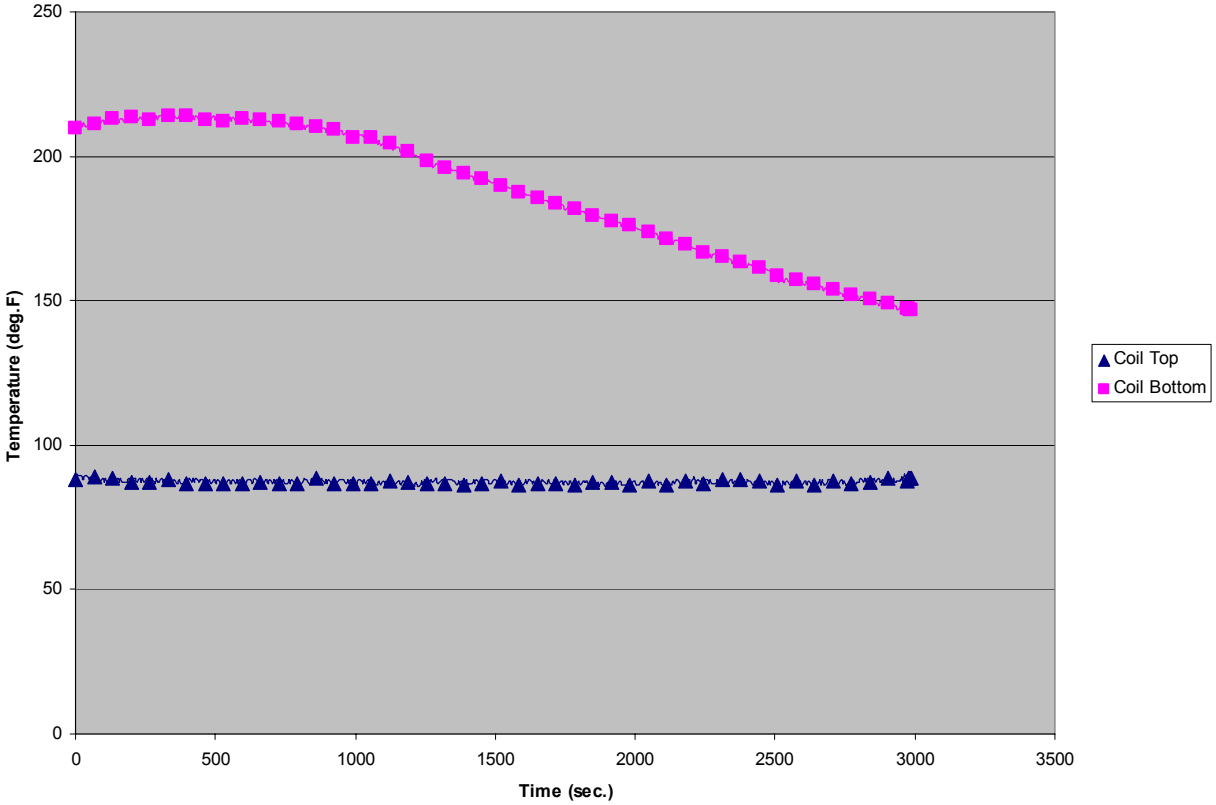


Figure 8. Temperature history at position 4 for Case 3.

Figure 8 gives the temperature at position 4 for Case 3. Figure 8 gives a temperature history very similar to that of Case 1 in Figure 5, yet the bottom temperature at time = 3000 sec. is higher in Figure 8. Case 3 has a larger width that that of Case 1. This may account for the higher final temperature at the coil bottom. It may be considered that a larger width coil may take longer to cool because air passing through would absorb more heat than it would for a smaller width. The temperature histories at positions 1, 3 and 4 for Case 3 are given in Appendix C. These positions were chosen to measure temperatures close to the inner, along the middle and close to the outer radius.

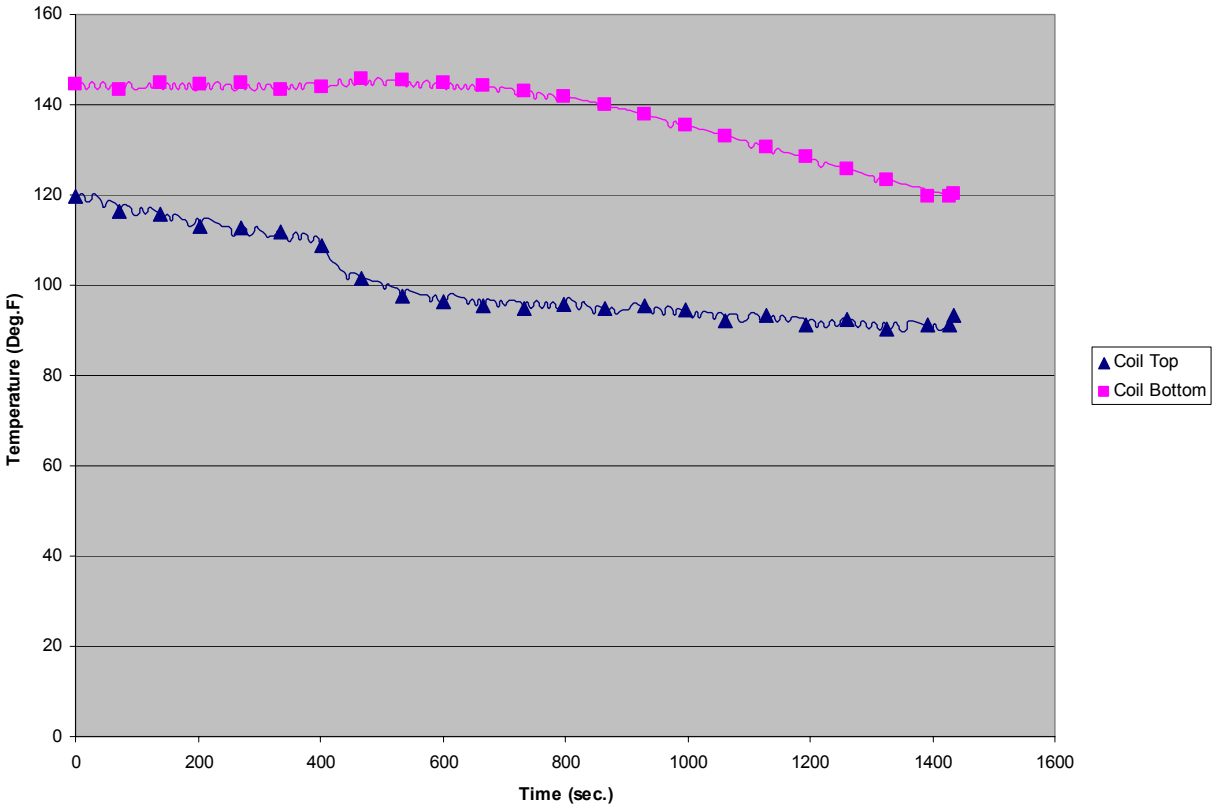


Figure 9. Temperature history at position 3 for Case 4.

Figure 9 gives the temperature history at position 3 for Case 4. Compared to the cases in the other figures, this coil had lower bottom and top initial temperatures. It was on the cooling bed for a shorter period of time. While the temperature histories for other cases were recorded for approximately 3000 seconds, the temperature history in this case was recorded for about 1400 seconds. This coil had the narrowest width of all the seven cases observed. The temperatures histories at positions 2 and 4 are approximately the same as that of Figure 9. The temperature history at the position 1 is similar, but starts at lower bottom and top initial temperatures. The histories at positions 1, 2, 3 and 4 for case 4 are given in Appendix D. Position

1 measured the temperatures near the inner edge, while positions 2, 3 and 4 measured for distances along the mid-radii.

The histories at positions 1, 3 and 5 for Case 5 are given in Appendix E. These positions measured temperatures near the inner, along the middle and close to the outer edge of the coil. In this case, data was recorded for a period of about 900 seconds. These histories slow little temperature change over their time. When compared, the other cases also show little change after 900 seconds. The histories at positions 1, 3, 4 and 6 for Case 6 are given in Appendix F. This case was the largest diameter coil. Position 1 measured the temperatures at the inner radius, positions 3 and 4 measured along the mid-radii and position 6 measured near the outer radius. The histories at positions 3, 4 and 6 give similar trends to that of Cases 1 and 3. The initial top temperatures are higher, however, they cool rapidly to approach the ambient temperature. The histories at positions 1, 3 and 5 for Case 7 are given in Appendix G. These positions measured the temperatures at the inner radius and along the mid-radii. Positions 3 and 5 are similar to that of Cases 1, 3 and 6, but the coil bottom temperatures drop more rapidly.

No conclusive observations could be made on the affects of individual parameters on the cooling of the coils. Further data acquisition of various size coils could be done in an attempt to verify tendencies and trends based on coil parameters. This study focuses on using the acquired data for comparison to any numerical modeling. However, observation of this data, could lead to the conclusion that coil width and initial temperature distribution have a major effect on the cooling of the coils.

5.4 Air Flow

Pitot tubes and inclined manometers were used to measure the difference in dynamic and static pressures at the bottom of a coil. A set of Pitot tubes where positioned at each of the six thermocouple positions along the cooling bed. The differences in pressures were measured at two times and three radial locations for a total of six differences in pressure readings for each of the seven cases. The locations chosen were near the inner, the middle and near the outer radius of a coil. This difference in fluid height was used in Bernoulli's equation [6] to determine the air velocity at the coil bottom using,

$$u = \sqrt{\frac{2gz(\rho_{oil} - \rho_{air})}{\rho_{air}}} \qquad (16)$$

where z is the difference in vertical column height between the dynamic and static pressures, ρ_{oil} is the density of the fluid in the inclined manometer and ρ_{air} is the density of the surrounding air. Each probe was position in close proximity to a thermocouple and only the air flow at the coil bottom was measured. For the numerical model, the air flow throughout the coil was assumed to be uniform. A value of z = 0.019 *ft*. (0.5 *mm*) was used in equation *(16)* was determined by averaging all the differences in pressure readings taken. Using equation *(16)*, a value of $u = 9.35\ {}^{ft}\!/_{\sec}$ ($2.85\ {}^{m}\!/_{\sec}$) was calculated for the air velocity throughout each coil.

6.0 COMPUTER RESULTS AND DISCUSSION

Comparisons of the field data and the numerical model were made in an attempt to verify the numerical results. Studies of the effects of the temperature distribution and input parameters were investigated.

6.1 Thermal Properties

The parameters in the energy balance equations used in the model are given in Table 2 for air [7, 8] and steel [9]. In the model, these values remained constant with temperature change.

Table 2. Parameters for Computer Model.

Parameter	U.S.	S.I.
ρ_a	$0.0624\ {}^{lb_m}\!/_{ft^3}$	$1.00\ {}^{kg}\!/_{m^3}$
c_a	$0.239\ {}^{Btu}\!/_{lb_m \cdot {}^{O}F}$	$1000\ {}^{J}\!/_{kg \cdot {}^{O}C}$
ρ_s	$490\ {}^{lb_m}\!/_{ft^3}$	$7850\ {}^{kg}\!/_{m^3}$
c_s	$0.104\ {}^{Btu}\!/_{lb_m \cdot {}^{O}F}$	$434\ {}^{J}\!/_{kg \cdot {}^{O}C}$
h	$4.40\ {}^{Btu}\!/_{hr \cdot ft^2 \cdot {}^{O}F}$	$25\ {}^{W}\!/_{m^2 \cdot {}^{O}C}$

6.2 Study of the Coil Cooling During Floor Placement

An analysis was performed to examine the temperature distribution that develops in the coil before it is placed on the cooling bed. The assumption is made that a coil is at a uniform temperature when it is removed from the furnace. It is then placed on the floor for a period of time, where the temperature in the coil changes before being placed on the cooling bed. The model used for the study was a 3.28 ft. (1.00 m) wide coil, with a gauge thickness of 0.0240 in. (0.609 m), 25 axial nodes for a time of 1000 seconds. The initial uniform temperature was 212 ^{O}F (100.0 ^{O}C). An air velocity of u = 0.0328 ${}^{ft}\!/_{\text{sec}}$ (0.01 ${}^{m}\!/_{\text{sec}}$) was used to simulate natural

convection of hot air rising up through the coil. At the top and sides, free convection was assumed to the ambient room temperature. The bottom was able to release heat by conduction to the metal floor. The value of the heat transfer coefficient at the top was $= 100\ {}^{W}\!/_{m^2 \cdot {}^{O}C}$ (17.6 ${}^{Btu}\!/_{hr \cdot ft^2 \cdot {}^{O}F}$) while the bottom, though primarily driven by conduction, but with some convection, was $h = 10\ {}^{W}\!/_{m^2 \cdot {}^{O}C}$ (1.76 ${}^{Btu}\!/_{hr \cdot ft^2 \cdot {}^{O}F}$). Figure 10 gives the temperature distribution across a coil width for a mid-radius location at different times. Details of the computer code used for this study are given in Appendix I.

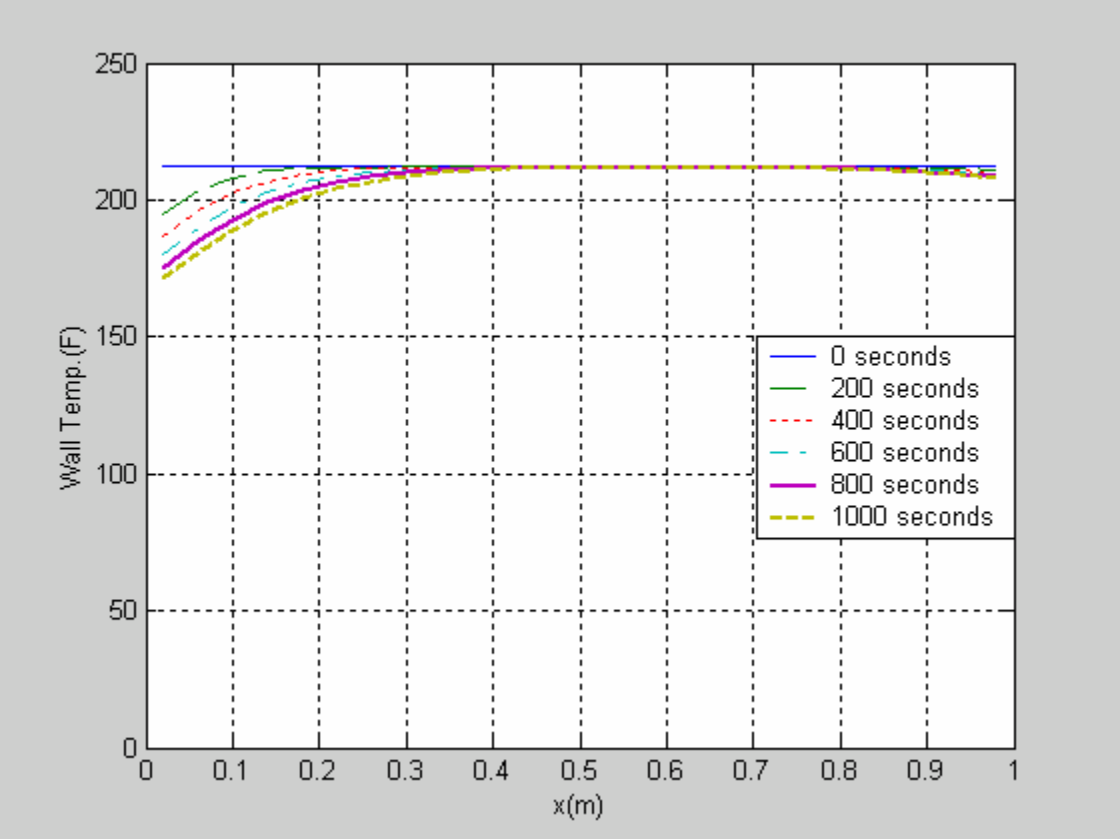


Figure 10. Temperature distribution across a coil width.

From Figure 10, the temperature versus axial location is given for 0, 200, 400, 600, 800 and 1000 seconds. With time, there is an increasing change in the temperature along the first quarter and last eighth of the coil width. The middle region has a uniform temperature. Since there was only experimental data for the top and bottom edges of the coil, there was no data to compare to the temperature distribution of Figure 10.

Figure 11 gives a sample plot of the initial temperature distribution used in the computer model. The distribution is for Case 1 which has a width of 49 in. (1.245 m) and a nodal spacing of 1 in. (25.4 mm). A linear distribution across the first quarter, up until approximately the twelfth node, of the coil is followed by a uniform temperature along the rest of its width.

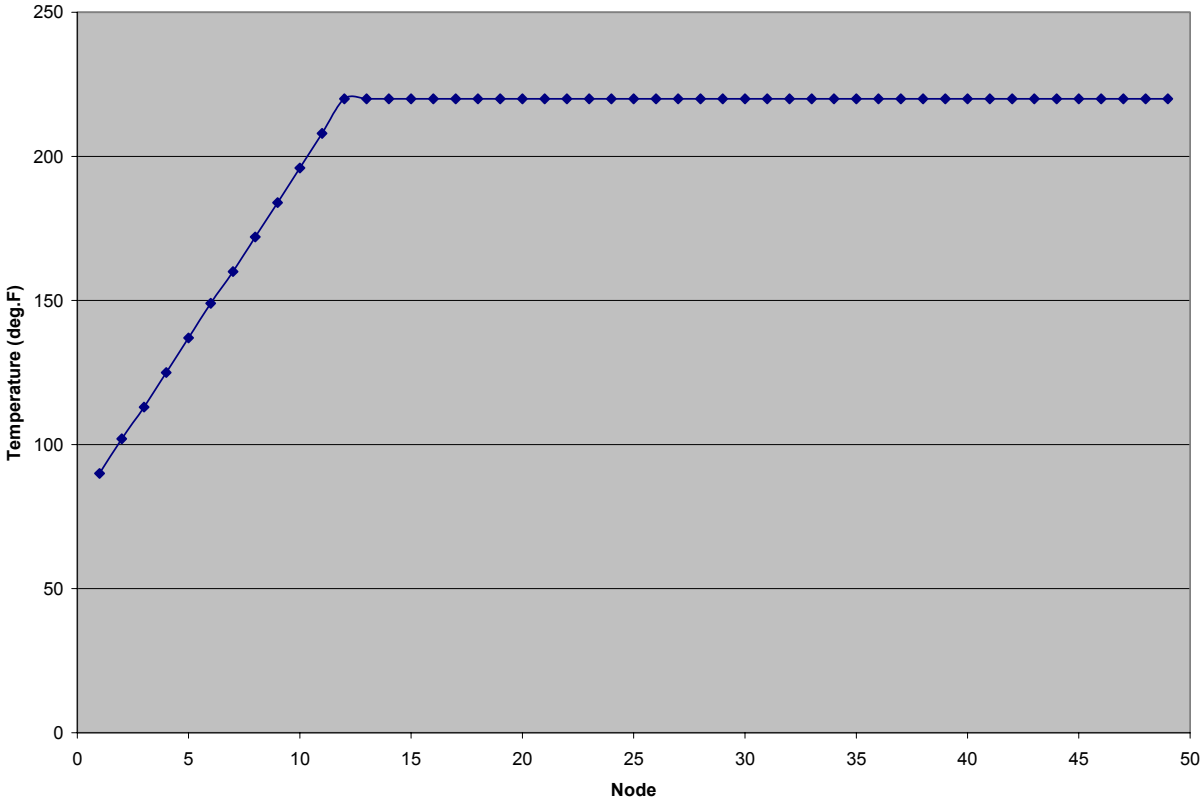


Figure 11. Initial temperature distribution for computer model.

6.3 Study of the Effect of the Convection Coefficient h

The convection coefficient, h, is a parameter that is difficult to measure and can have a range of values even for similar setups. Since an exact value of h is not known for the problem here,

several values were studied to see the effect of varying h on the temperature history. The input data and parameters for Case 3 were used. An air velocity of u = 2.46 $^{ft}/_{\text{sec}}$ (0.75 $^{m}/_{\text{sec}}$) was used. The gauge thickness was 0.0240 in. (0.609 mm), the coil width was 61 in. (1.538 m) and the outer diameter was 95 in. (2.413 m). The initial coil top temperature was 90 ^{O}F (32.22 ^{O}C) the bottom was 210 ^{O}F (98.89 ^{O}C) and the ambient air temperature was 87.6 ^{O}F (30.89 ^{O}C). One node per inch of width was used in the model. Heat transfer coefficient values of h = 2.5, 25, 250 and 2500 $^{W}/_{m^2 \cdot \deg.C}$ (4.4, 44, 440 and 4400 $^{Btu}/_{hr \cdot ft^2 \cdot \deg.F}$) were used in the computer model and the temperature history at the top and bottom of the middle of the coil was calculated. Free convection to the ambient air was assumed on the sides of the coil. Ambient air was drawn into the top of the coil by forced convection. Air, with stored heat that was released to it by the coil, was drawn out of the coil bottom by forced convection. The temperature histories are given in Figure 12.

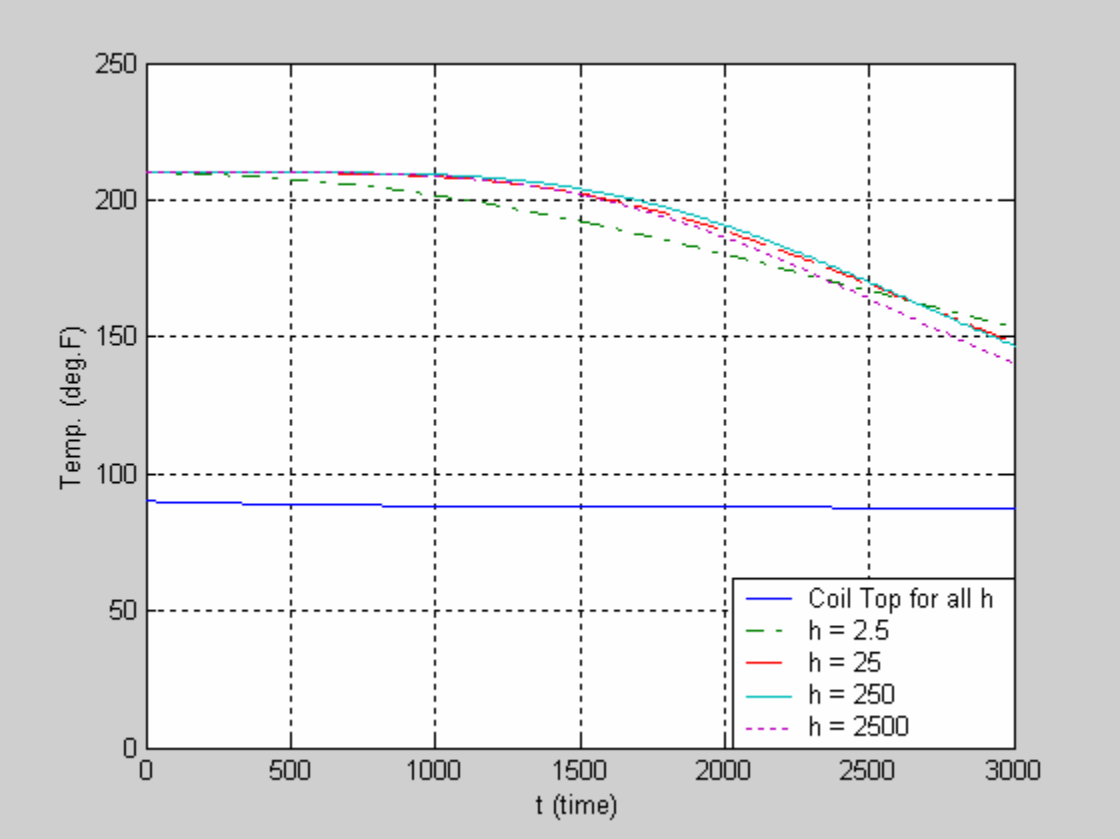


Figure 12. Temperature history for varying heat transfer coefficient at position 4 for Case 3.

From the data in Figure 12, it can be seen that there are only minor differences in the results for different h values. The figure shows that using a lower h value (2.5) results in a more rapid descend in the temperatures, yet a higher coil bottom temperature at the end. For an h value of 2500 $W/m^2 \cdot \deg.C$ (4400 $Btu/hr \cdot ft^2 \cdot \deg.F$), there is a slighter higher temperature drop at the end. For h values of 25 $W/m^2 \cdot \deg.C$ (44 $Btu/hr \cdot ft^2 \cdot \deg.F$) and 250 $W/m^2 \cdot \deg.C$ (440 $Btu/hr \cdot ft^2 \cdot \deg.F$), there seems to be a very similar curve at the beginning and end. Since the

value of h showed only minor differences, the same value cold be used for the top, bottom and side boundaries.

From the results of this study it was seen that the value of h has a small effect on the temperature history. For this reason a value of $h = 25\ {}^{W}\!/\!_{m^2 \cdot \deg.C}$ ($4.40\ {}^{Btu}\!/\!_{hr \cdot ft^2 \cdot \deg.F}$) was used for the remainder of this study and is a value that is in the range of those for air flow over steel [7].

6.4 Study of the Effect of Air Velocity

The appropriate air velocity for the model was obtained by varying its value until a reasonable fit of the experimental temperature history was achieved. The results using the measured air velocity did not give a good fit of the experimental data. Using the measured value for the air velocity, $u = 0.0328\ {}^{ft}\!/\!_{\text{sec}}$ ($0.01\ {}^{m}\!/\!_{\text{sec}}$) caused the coil to take much longer to cool than that of the experimental results showed. Values for the velocity that gave a close curve fit ranged from $u = 1.64$ (0.50) to $3.28\ {}^{ft}\!/\!_{\text{sec}}$ ($1.00\ {}^{m}\!/\!_{\text{sec}}$). The air velocity of $u = 2.46\ {}^{ft}\!/\!_{\text{sec}}$ ($0.75\ {}^{m}\!/\!_{\text{sec}}$) was used for most of the cases in the model. The field data for Case 1 was used for comparison. An air velocity of $u = 2.46\ {}^{ft}\!/\!_{\text{sec}}$ ($0.75\ {}^{m}\!/\!_{\text{sec}}$) was used, for a gauge thickness of 0.0201 in. (0.510 mm) and coil width of 49 in. (1.245 m). The outer diameter was 101.5 in. (2.578 m) along with an initial coil top temperature of 98 ^{O}F (36.67 ^{O}C), 230 ^{O}F (110.0 ^{O}C) for the bottom and an ambient air temperature of 89.8 ^{O}F (32.11 ^{O}C). One node per inch was used for the 49 in. (1.245 m) wide coil. The value of the heat transfer coefficient used was $h = 25$

$W/m^2 \cdot \deg.C$ (4.40 $Btu/hr \cdot ft^2 \cdot \deg.F$). Free convection to the ambient air was assumed on the sides of the coil. Ambient air was drawn into the top of the coil by forced convection. Air, with stored heat that was released to it by the coil, was drawn out of the coil bottom by forced convection. The results for a range of air velocities for this case are given in Figure 13.

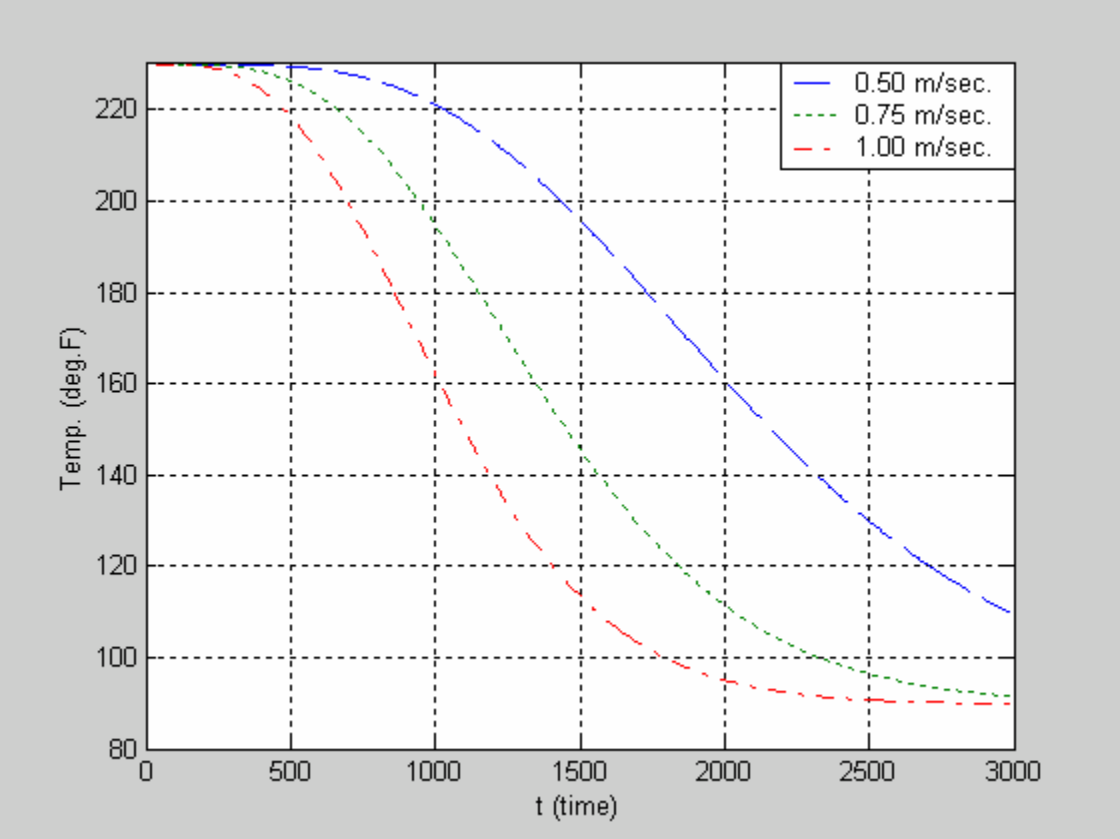


Figure 13. Temperature history for varying air velocities.

From Figure 13, the range of these velocities gave a temperature history similar to that of Case 1, but a velocity of $u = 2.46$ ft/sec (0.75 m/sec) gave the closest approximation in most of

the cases. The exceptions were Case 2, which used a velocity of 1.64 ft/sec (0.50 m/sec), and Case 4, which used a velocity of u = 3.28 ft/sec (1.00 m/sec). Values lower than u = 1.64 ft/sec (0.50 m/sec) showed minor decrease over the temperature history, while values higher than u = 3.28 ft/sec (1.00 m/sec) showed rapid decrease.

6.5 Study of the Effect of Conduction

The effect of heat conduction along the width of a coil during cooling was considered. The energy balance equations for air and steel were modified to include heat conduction terms [10] and the computer model was modified to include these terms. The same parameters were used as that of the model of the coil on the floor. Details of this computer code are listed in Appendix J. Figure 14 shows the history of the top and bottom coil temperatures of an individual steel wrap for input data considering convection only and both conduction and convection.

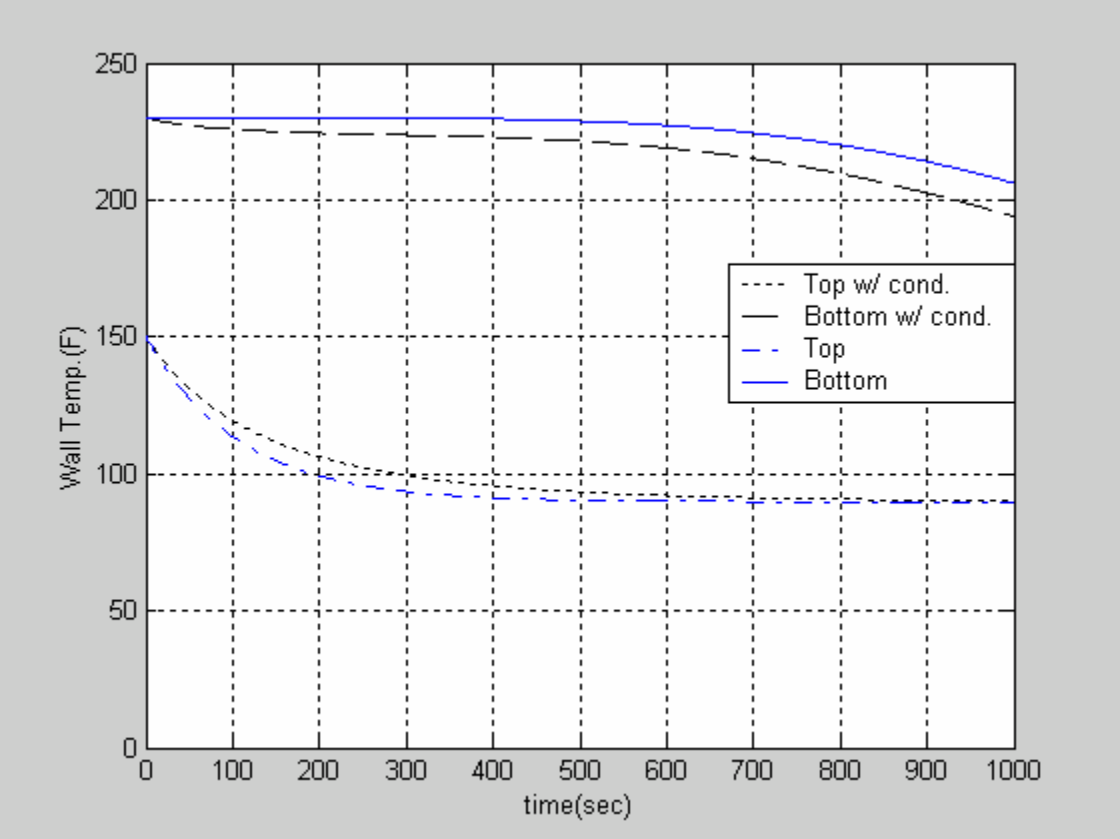


Figure 14. Temperature history for conduction vs. nonconduction.

The top temperature is higher with axial conduction because there is heat flow from the interior of the coil by conduction. The maximum difference between the coil top for the two cases is about 10 degrees. The reverse is true for the bottom of the coil. Without axial conduction, the bottom temperature is higher because there is no heat flow, via axial conduction, toward the top of the coil. With conduction, the coil bottom is releasing heat by both conduction and convection. After about 1000 seconds of cooling, the maximum difference between conduction and convection temperatures is about 15 degrees, which is at the coil bottom. This study verified the original hypothesis that most of the heat transfer is due to axial convection and not axial conduction.

6.6 Study of the Temperature Distribution Across the Coil Radius

The computer model was used to examine the temperature distribution across the coil radius at various times. The output was modified so that the temperature profile at various times along the first 10 wraps of the coil could be examined. The field data for Case 1 was used for comparison. An air velocity of $u = 2.46\ {}^{ft}\!/_{\sec}$ ($0.75\ {}^{m}\!/_{\sec}$) was used. The gauge thickness was 0.0201 in. (0.510 mm), the coil width was 49 in. (1.245 m) and the outer diameter was 101.5 in. (2.578 m). The initial coil top temperature was 98 ^{O}F (36.67 ^{O}C) and 230 ^{O}F (110.0 ^{O}C) for the bottom and an ambient air temperature was 89.8 ^{O}F (32.11 ^{O}C). One node per inch was used for the 49 in. (1.245 m) wide coil. This temperature history was of a mid-width node. Free convection to the ambient air was assumed on the sides of the coil. Ambient air was drawn into the top of the coil by forced convection. Air, with stored heat that was released to it by the coil, was drawn out of the coil bottom by forced convection. Figure 15 shows this temperature distribution at different times.

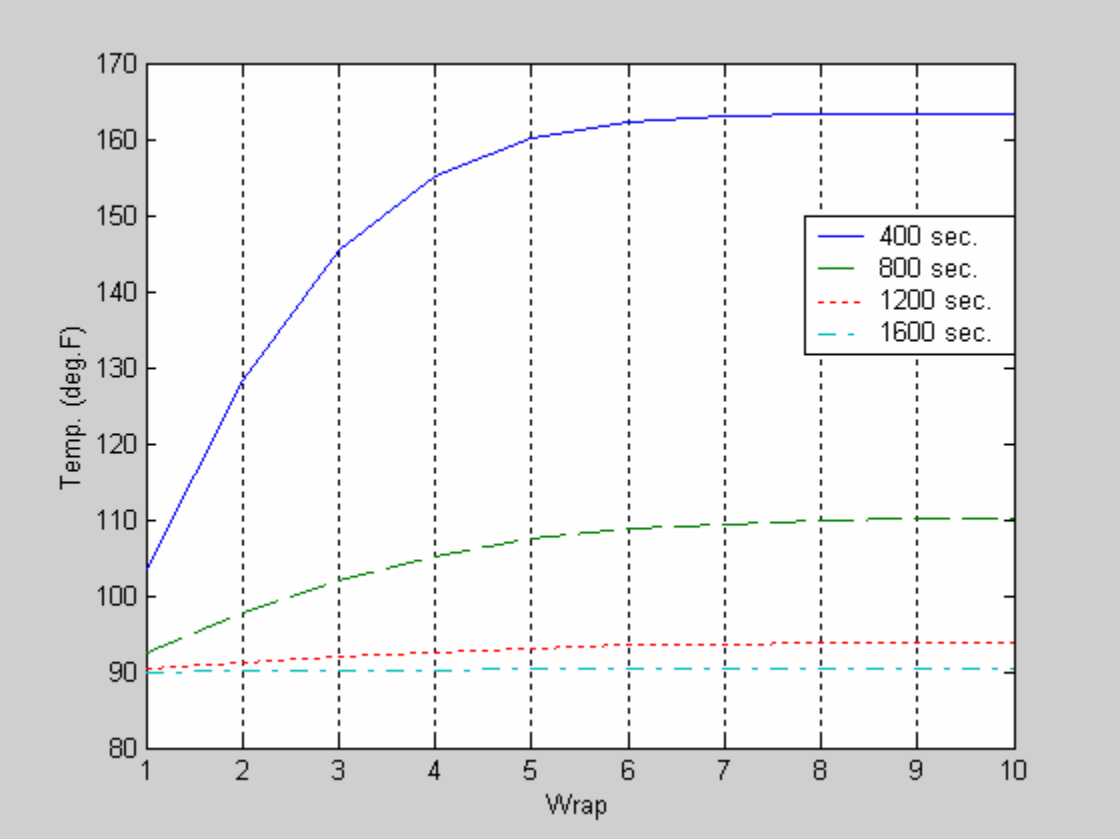


Figure 15. Temperature distribution across the coil radius for Case 1.

From the curves in Figure 15, it can be observed that there is a temperature profile across the first seven wraps after which there is a uniform distribution. The model also showed that there is a similar distribution for the last seven wraps of the model. At a given time for a 300 wrap model, this would mean that wraps 1 through 7 and 293 through 300 would show this profile, while the rest of the wraps would be at approximately the same temperature. The temperatures along the top and bottom of the coil show similar distribution across their widths.

6.7 Comparison of the Computer vs. Experimental Results

The results of the computer model were compared to the experimental data. An air velocity of u = 2.46 ft/sec (0.75 m/sec) and a convection heat transfer coefficient of h = 4.40 $Btu/hr \cdot ft^2 \cdot {}^{O}F$ (25 $W/m^2 \cdot {}^{O}C$) were used. From Table 2, the air density and specific heat used were ρ_a = 0.0624 lb_m/ft^3 (1.00 kg/m^3) and c_a = 0.239 $Btu/lb_m \cdot {}^{O}F$ (1000 $J/kg \cdot {}^{O}C$), respectively. The steel density and specific heat used were ρ_s = 490 lb_m/ft^3 (7850 kg/m^3) and c_s = 0.104 $Btu/lb_m \cdot {}^{O}F$ (434 $J/kg \cdot {}^{O}C$), respectively. Figure 16 gives a temperature history plot comparing the results of the computer model to position 1 in Case 1.

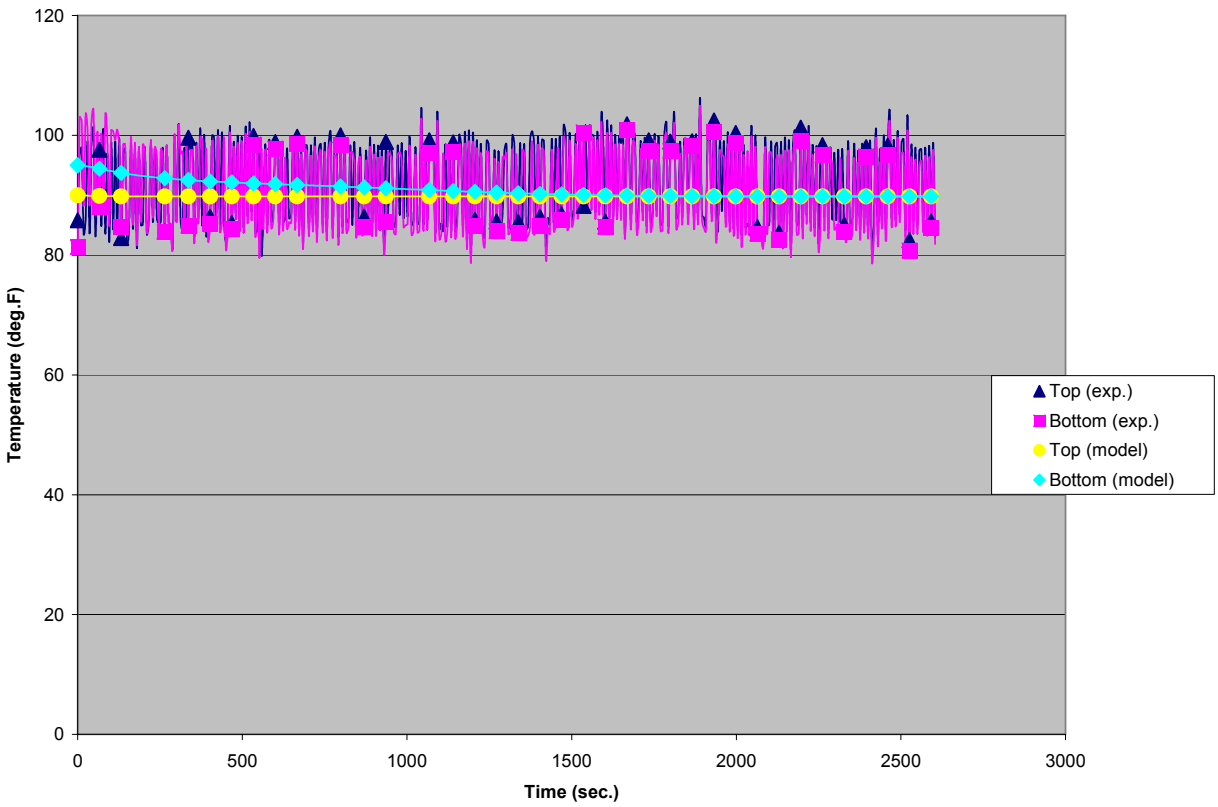


Figure 16. Comparison of computer model to position 1 in Case 1.

From Figure16, it is observed that the top and bottom temperatures are near the ambient through the recording time as with the history of position 1 for Case 1 in Figure 5. The bottom temperature starts a little higher, but quickly approaches the ambient, compared to the data in Figure 5 which shows temperatures fluctuations along the ambient. Since the inner and outer wraps of this coil were exposed to the ambient air for a period of time before being placed on the cooling bed, each wrap had time to approach the ambient temperature. Because of this condition, the histories in Figures 5 and 16 do not give much information of interest on the cooling behavior of the coil.

Figure 16 shows a typical temperature history of the inner radius position for most of the seven cases from the computer model. The exceptions were Cases 2, 3 and 7. In Case 2, no experimental data was taken at position 1 because the coil was not placed directly centered on the cooling bed. As a result, the thermocouple at position 1 was not touching the inner radius of the coil. Instead, position 2 was the inner most position used for Case 2. In Case 3, the experimental data at position 1 did not agree with the results of the model. A possibility may be that the coil bottom did not cool to the ambient temperature in the time that it was sitting on the floor before being placed on the cooling bed. Position 1 in Case 7 has a temperature history similar to that of position 1 in Case 3.

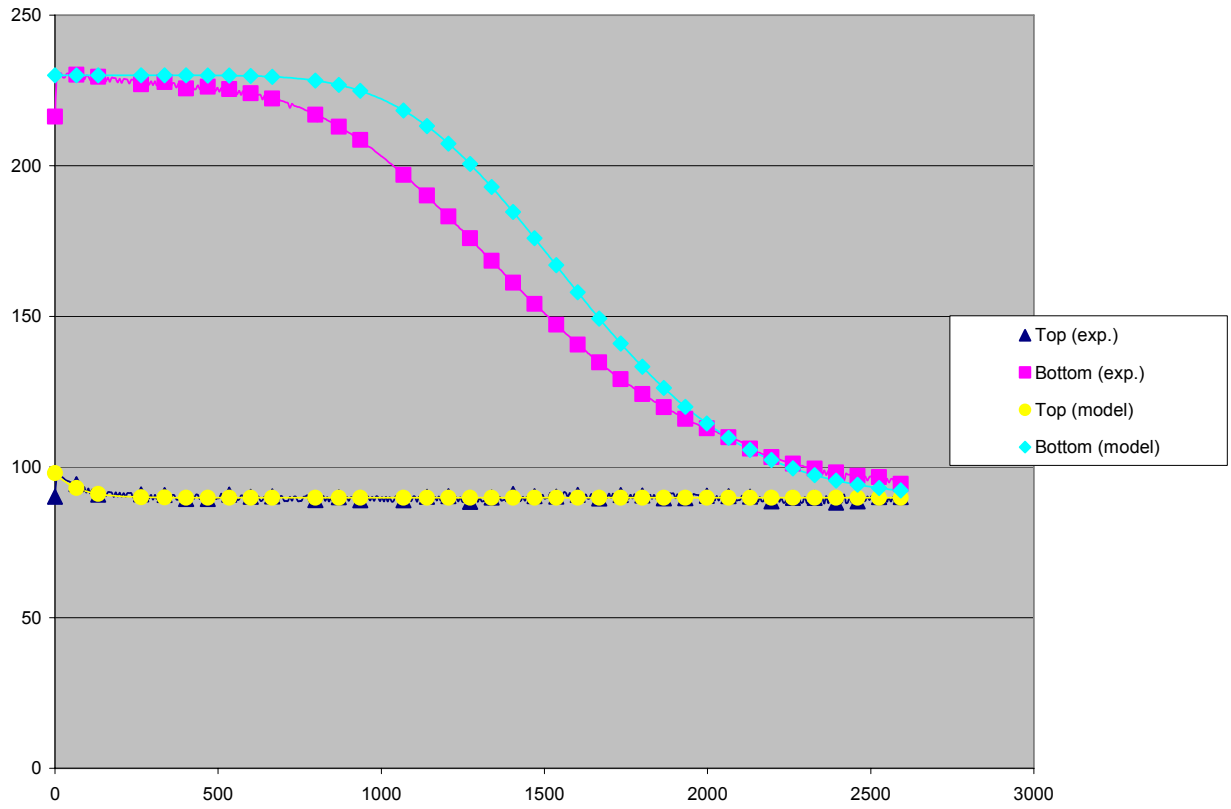


Figure 17. Comparison of computer model to position 5 in Case 1.

Figure 17 gives the results for a mid-radius wrap from the input data used in Case 1. This can be compared to position 5 for Case 1 in Figure 6. From Figure 17, the computer model gives a fairly constant temperature over the first 500 seconds of cooling for the coil bottom, compared to Figure 6 which shows a rise followed by a drop in temperature over the same time. A possible reason for this could be a temperature, somewhere along the mid-width, that is higher than that of the bottom. The upstream air at that point would cause heat released to it to temporarily raise the temperatures of the steel near the coil bottom. This is an indication that the initial temperature distribution of Figure 13 would give a more correct initial temperature distribution

than that of Figure 14. Since no experimental data of the mid-width temperatures were taken, the exact temperature profile along the width of the coil could not be determined. The coil top temperature gives a curve close to that of Figure 6. Comparing these results, the coil top approached the ambient air temperature at the same rate for both the experimental data and computer model.

Figure 17 gives a typical temperature history for mid-radius positions from the computer model. The exceptions were Cases 3, 4 and 5. In Case 3, positions 3 and 4 do not have the same temperature histories. In Case 4, the temperature history for the experimental data was recorded for approximately one half of the time of the other cases. In Case 5, positions 3 and 5 for the experimental data show temperatures that do not decrease significantly over time.

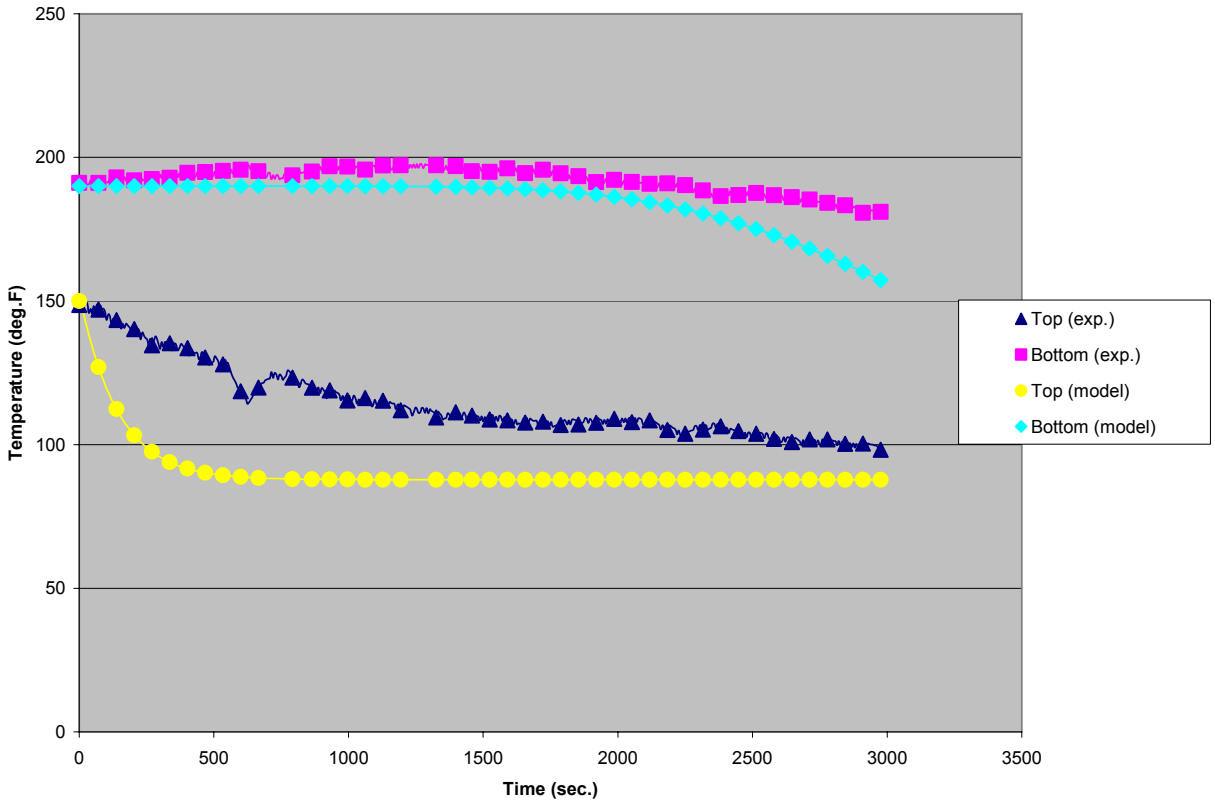


Figure 18. Comparison of computer model to position 4 in Case 2.

Figure 18 gives results compared to that of position 4 for Case 2 in Figure 7. Compared to Figure 7, the coil bottom temperature in Figure 18 stays constant throughout the entire cooling time. Figure 7 shows the coil bottom temperature rise then fall. Again the possibility of temperatures somewhere along the mid-width are considered higher than that of the bottom temperatures are responsible for this rise. The computer model shows a negligible change in the coil bottom temperature over the period of cooling.

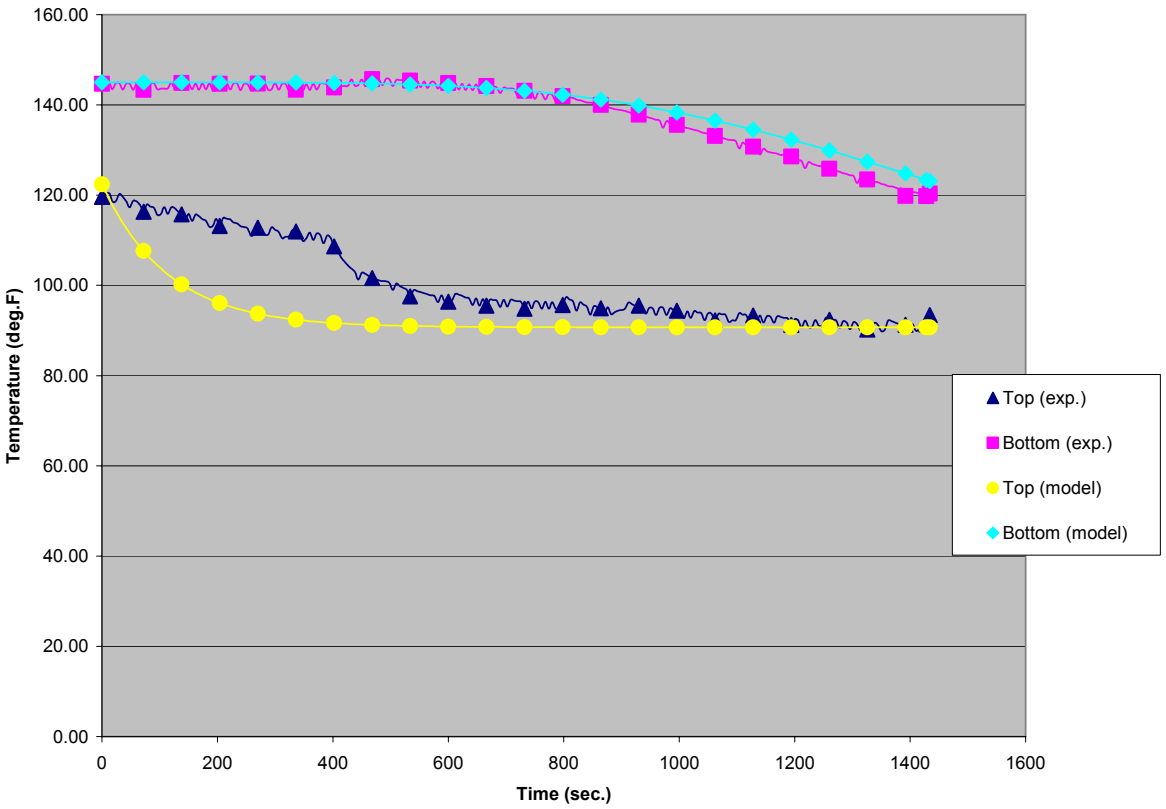


Figure 19. Comparison of computer model to position 3 in Case 4.

Figure 19 gives results compared to that of position 3 for Case 4 in Figure 9. Compared to Figure 9, Figure 19 gives very close approximation. The top and bottom coil temperatures in both figures show very similar histories. Case 4 was recorded for a much shorter period of time than the other cases. There is no noticeable rise in temperature at the beginning of the curve. The top and bottom temperatures appear to decrease at the same rates in both figures. Compared to the other cases, this coil has a much lower initial bottom temperature. Case 4 is the narrowest width of all the cases. It may be considered that the width my affect the overall temperature difference of the top and bottom edges of the coil, yet there was no conclusive evidence.

7.0 CONCLUSION

A computer model was created to analyze the cooling of an open steel coil. This model was compared to field data to determine its validity. Overall, the computer model gave a close approximation of the temperature history of the top and bottom edges along the mid-radius of a coil. Inner coil temperatures could not be measured in any feasible manner because of the coil geometry. It would be difficult to position thermocouples within the coil, because of the small spacing, in order to measure these inner temperatures. Observation of the experimental data would suggest the possibility of temperatures higher than that of the coil bottom within the coil width. The computer model assumed a temperature distribution throughout the coil width, but the calculations determined the highest temperatures to be at the coil bottom. Further experimental testing could be performed to determine the temperature distributions throughout the widths of the coils. Data from these tests could be used to modify and possibly improve on the computer model.

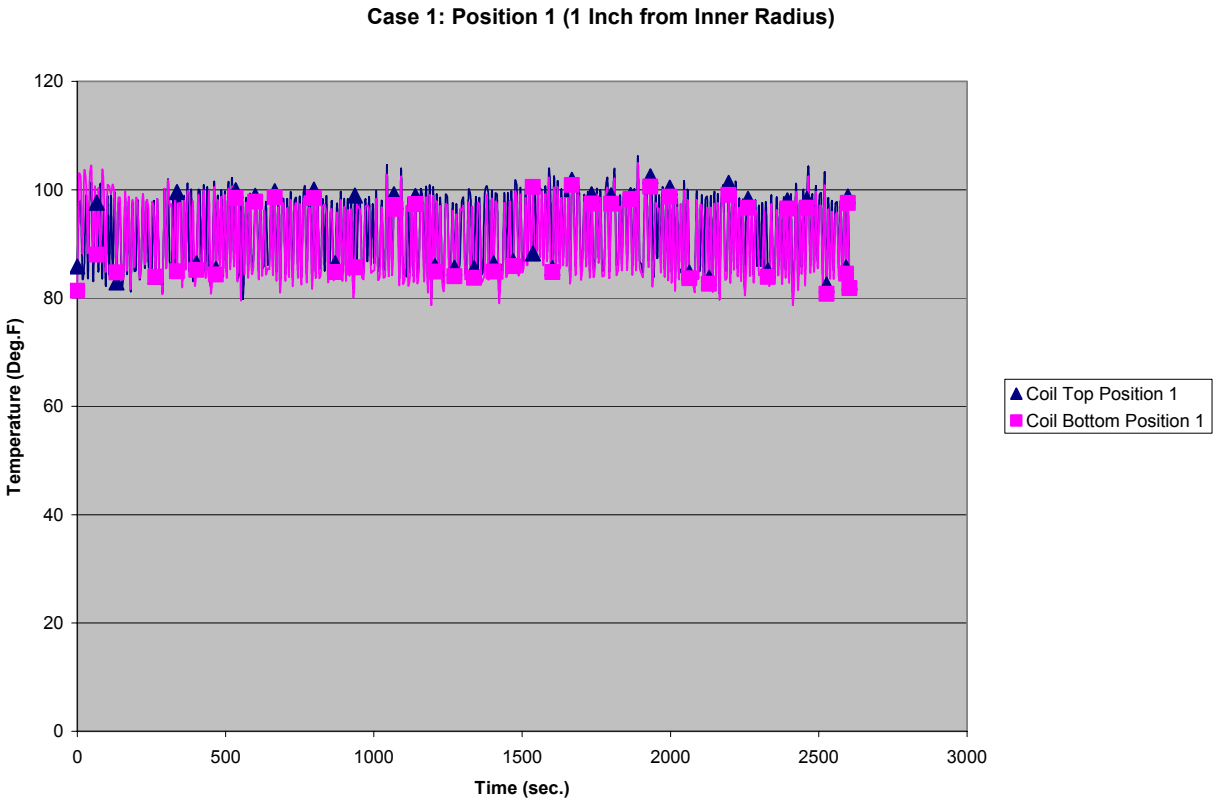
When compared to the seven experimental cases, using values of 25 $W/m^2 - \deg.C$ (4.40 $Btu/hr - ft^2 - \deg.F$) for h, 2.46 $ft/\sec$ (0.75 $m/\sec$) for u and the intial temperature distribution of Figure 13 gave a reasonably accurate temperature history of the temperatures for the top and bottom edges for all seven coil cases. Knowing the top and bottom temperatures, along with the

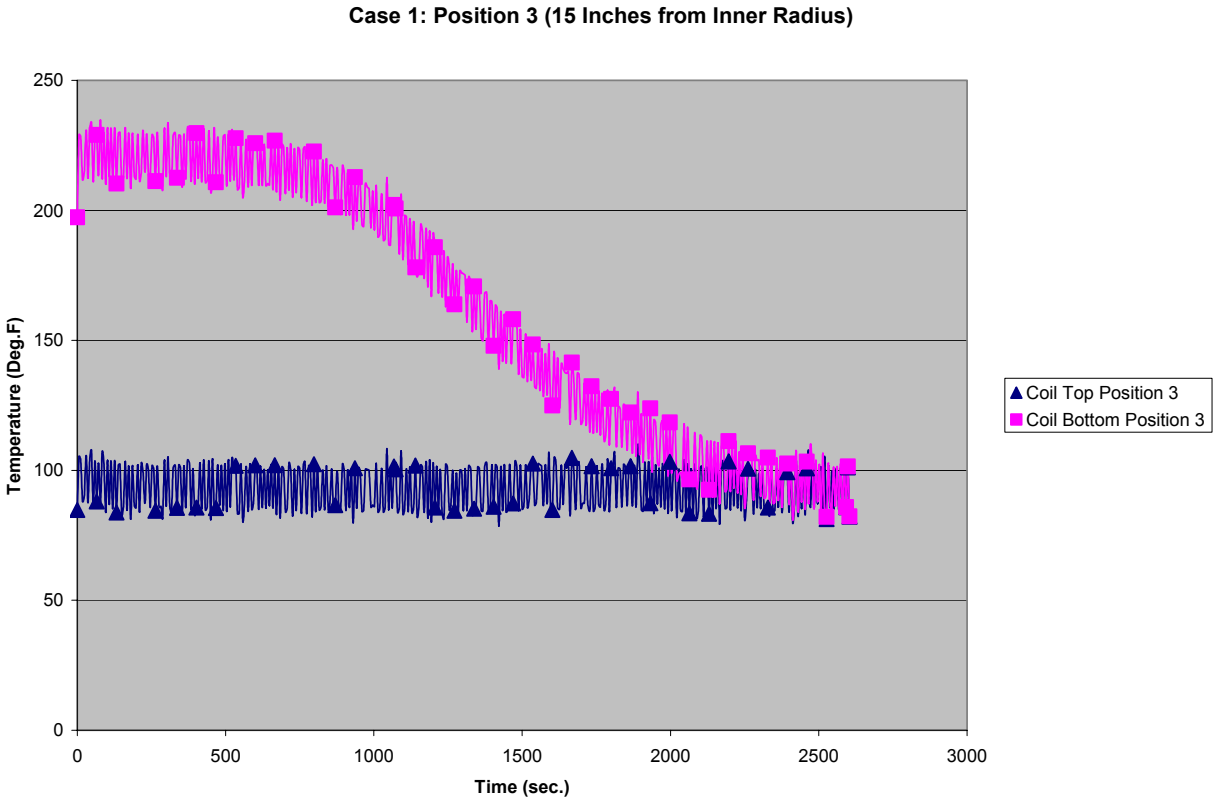
other input data for a coil, as its placed on a cooling bed, these parameters should give an accurate temperature history prediction.

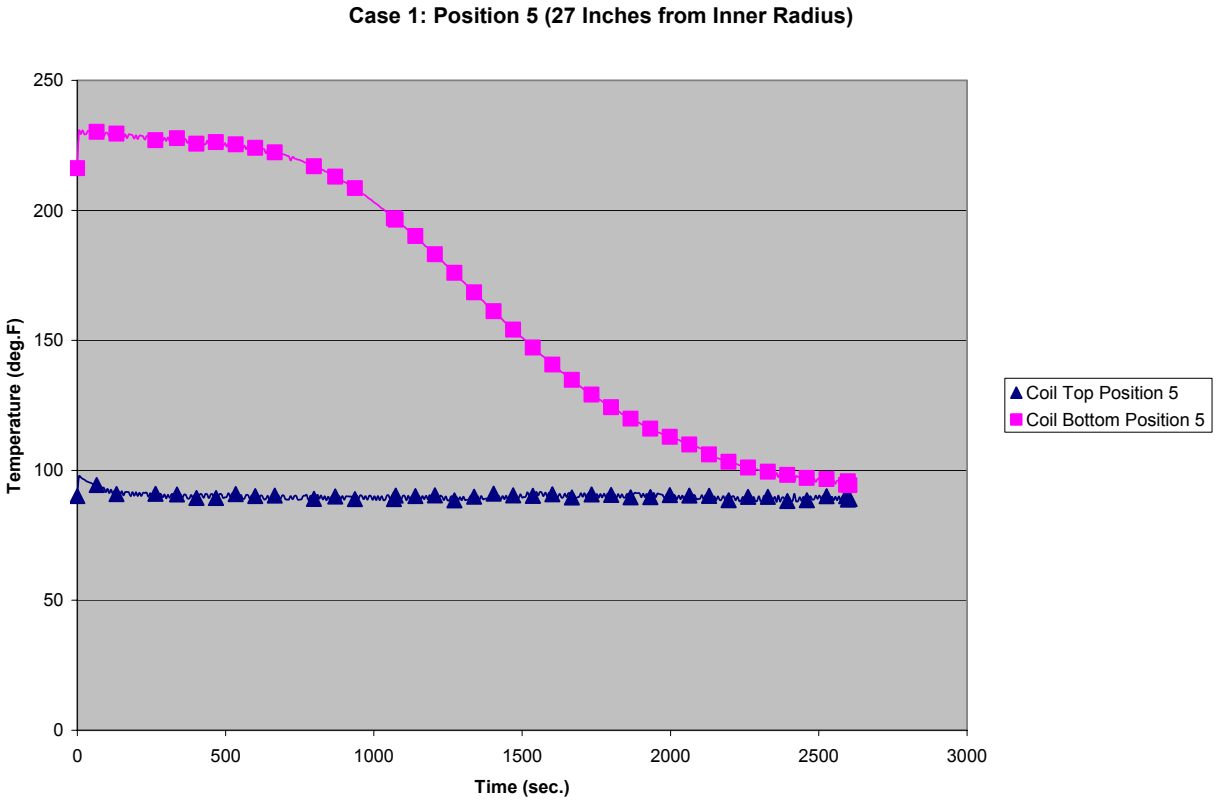
APPENDICES

APPENDIX A

Experimental Data for Case 1

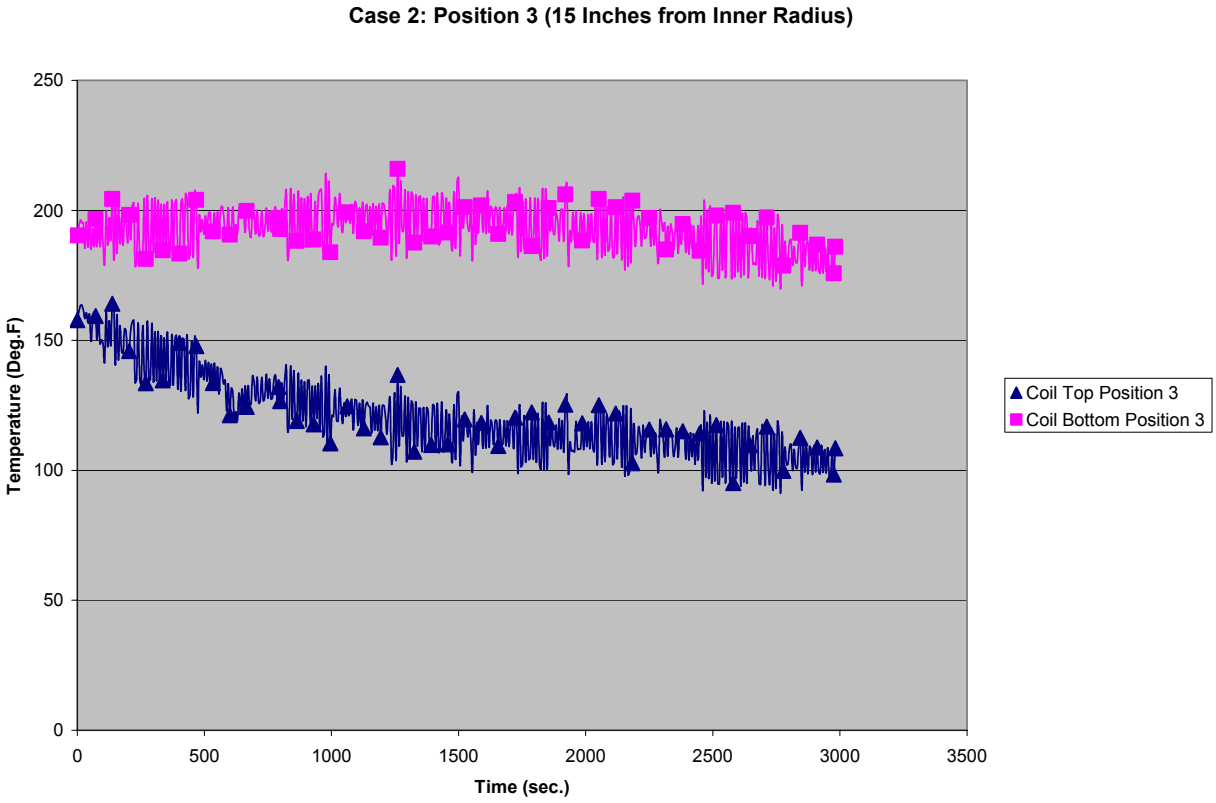



Case 1: Position 3 (15 Inches from Inner Radius)
250
200
150
100
50
0
Temperature (Deg.F)
0
500
1000
1500
2000
2500
3000
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3

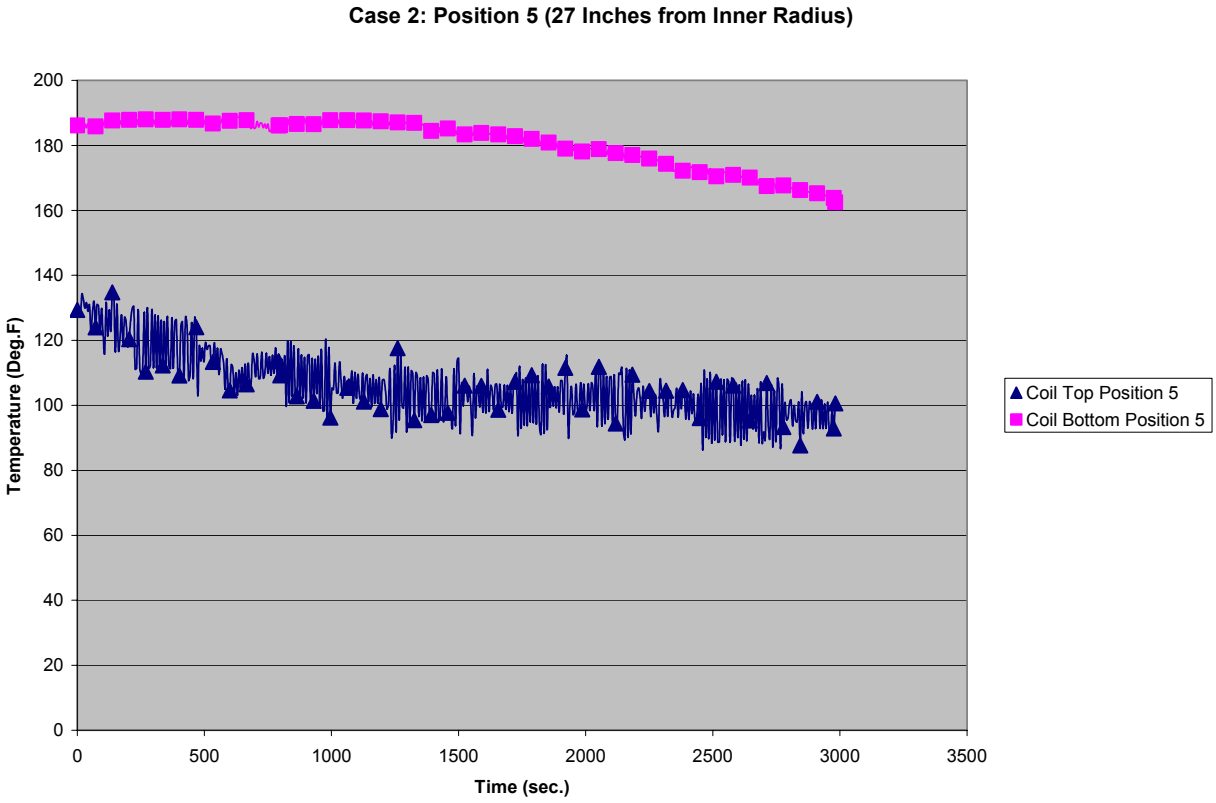

Case 1: Position 5 (27 Inches from Inner Radius)
250
200
150
100
50
0
Temperature (deg.F)
0
500
1000
1500
2000
2500
3000
Time (sec.)
Coil Top Position 5
Coil Bottom Position 5

APPENDIX B

Experimental Data for Case 2

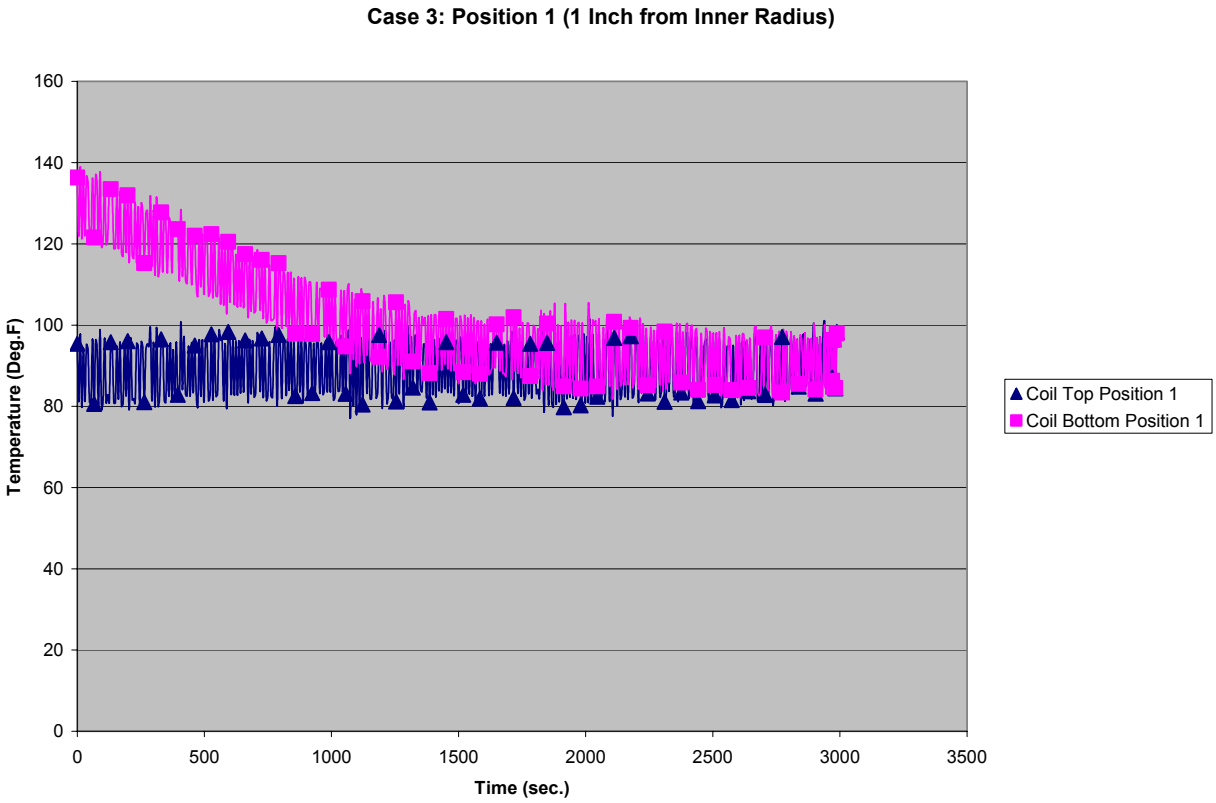

Case 2: Position 3 (15 Inches from Inner Radius)
250
200
150
100
50
0
Temperature (Deg.F)
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3

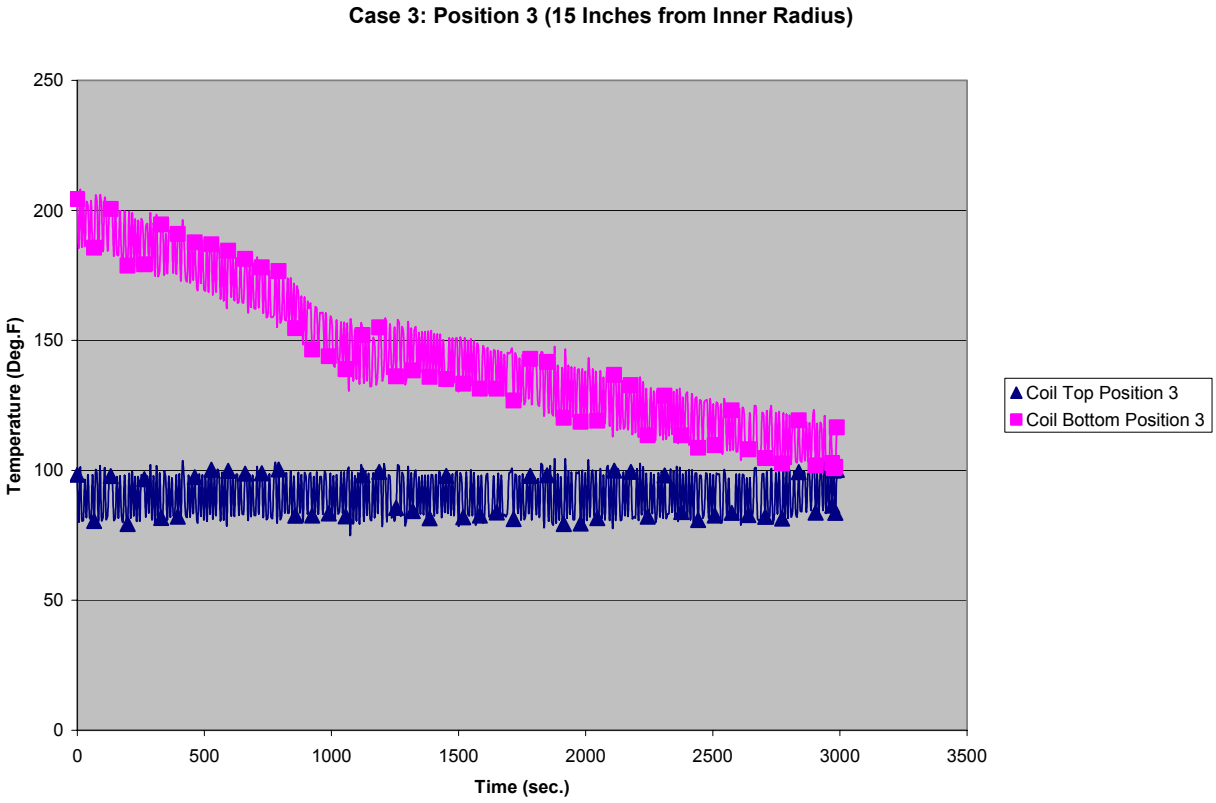

Case 2: Position 4 (21 Inches from Inner Radius)
Temperature (Deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 4
Coil Bottom Position 4

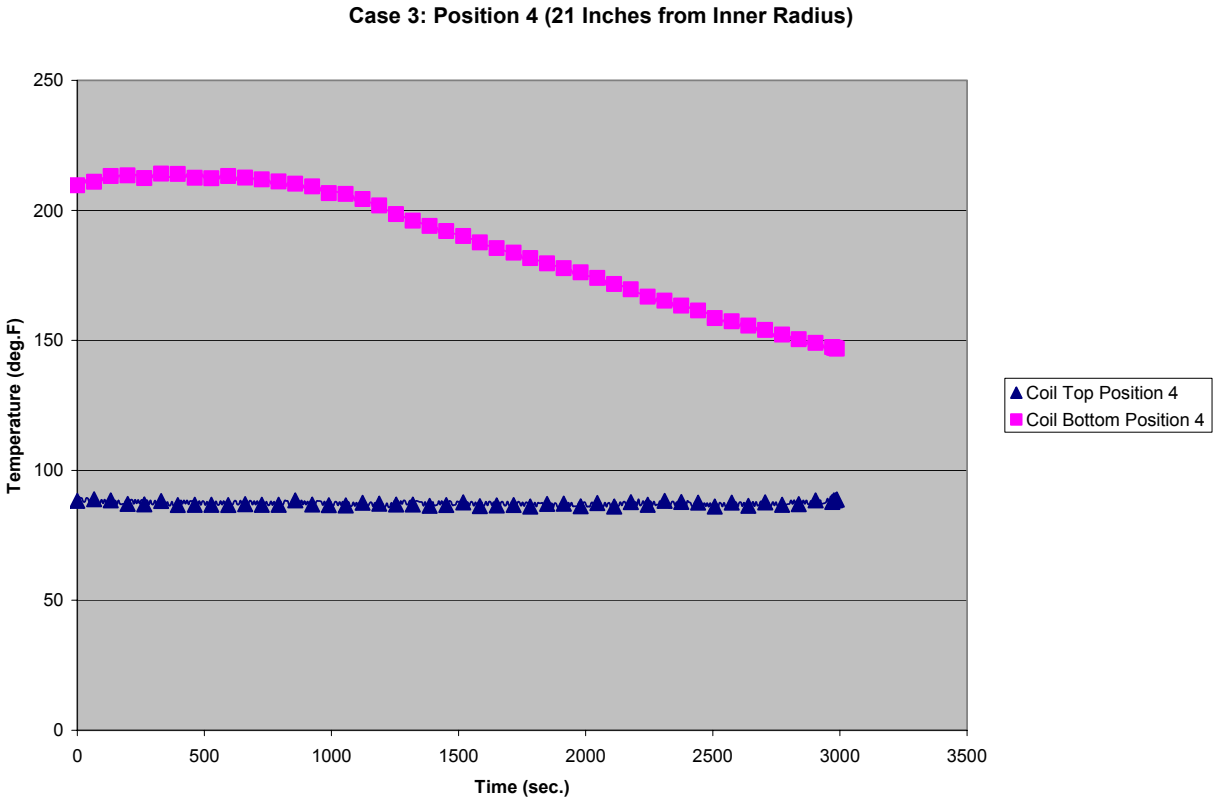

Case 2: Position 5 (27 Inches from Inner Radius)
Temperature (Deg.F)
200
180
160
140
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 5
Coil Bottom Position 5

APPENDIX C

Experimental Data for Case 3

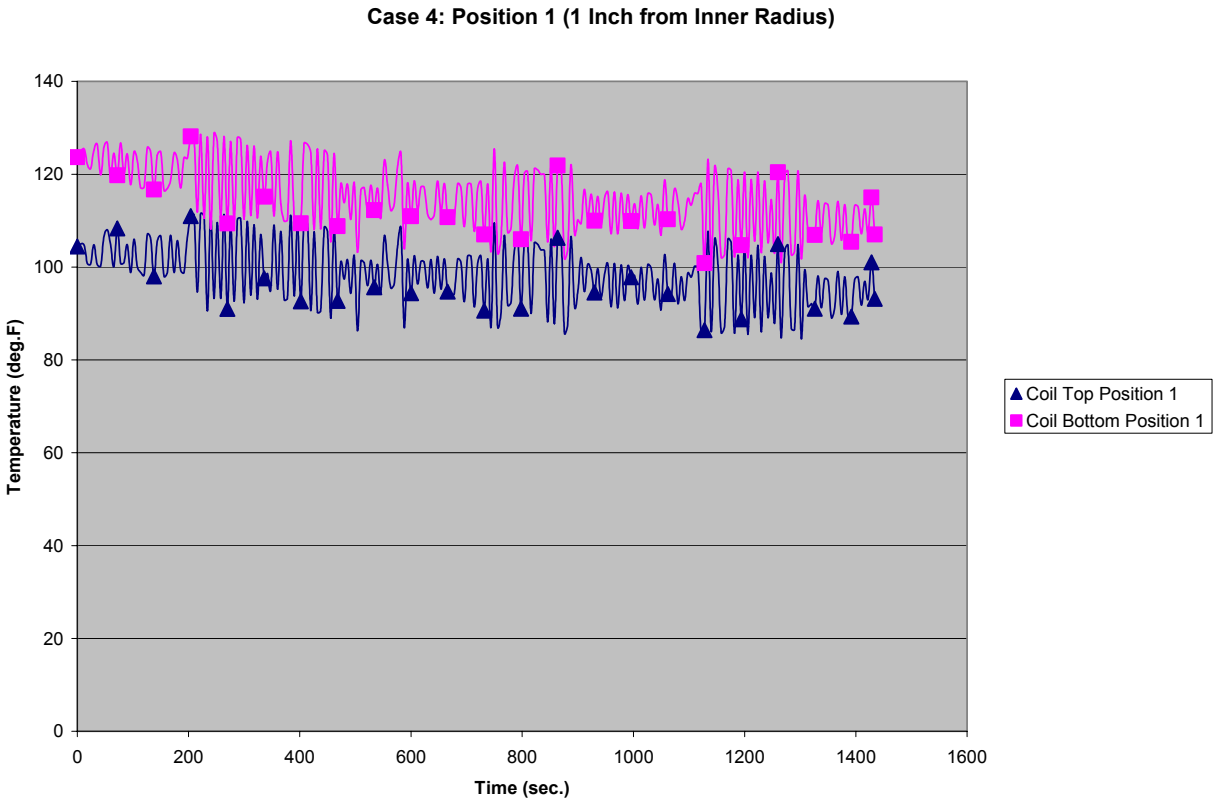


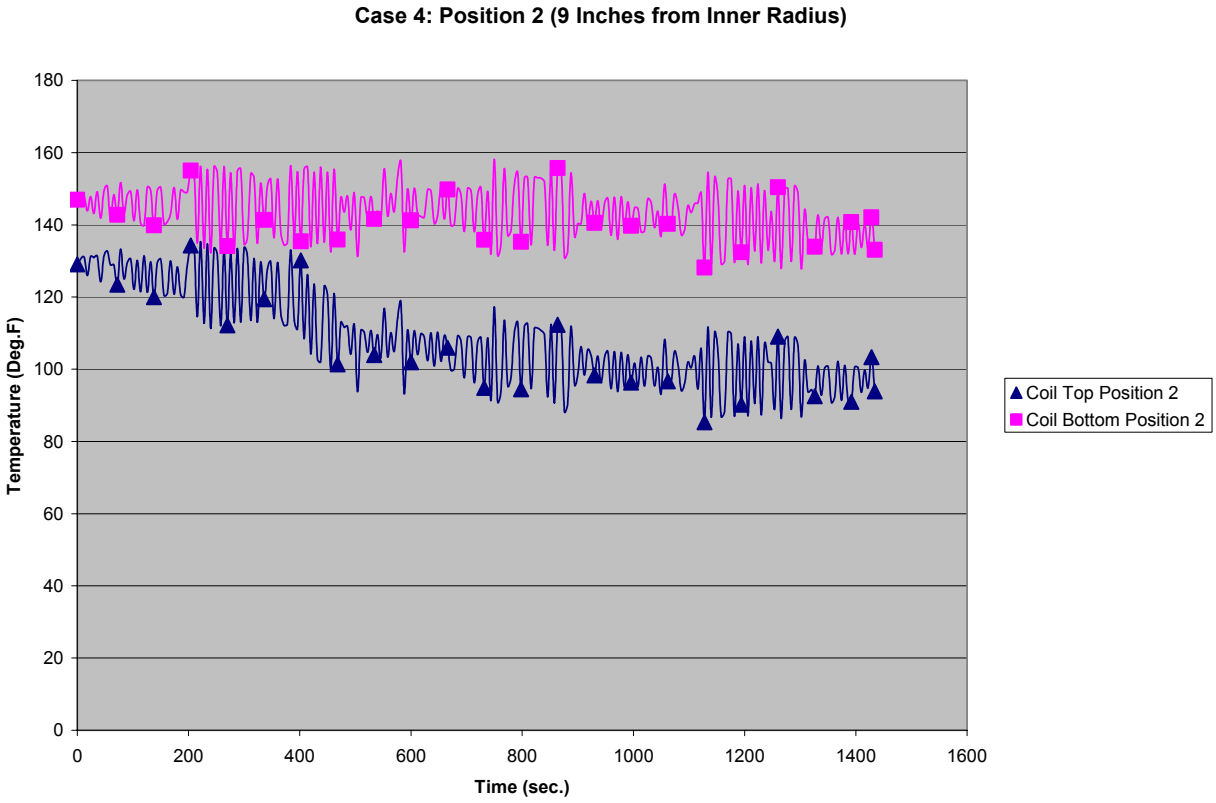

Case 3: Position 3 (15 Inches from Inner Radius)
250
200
150
100
50
0
Temperature (Deg.F)
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3

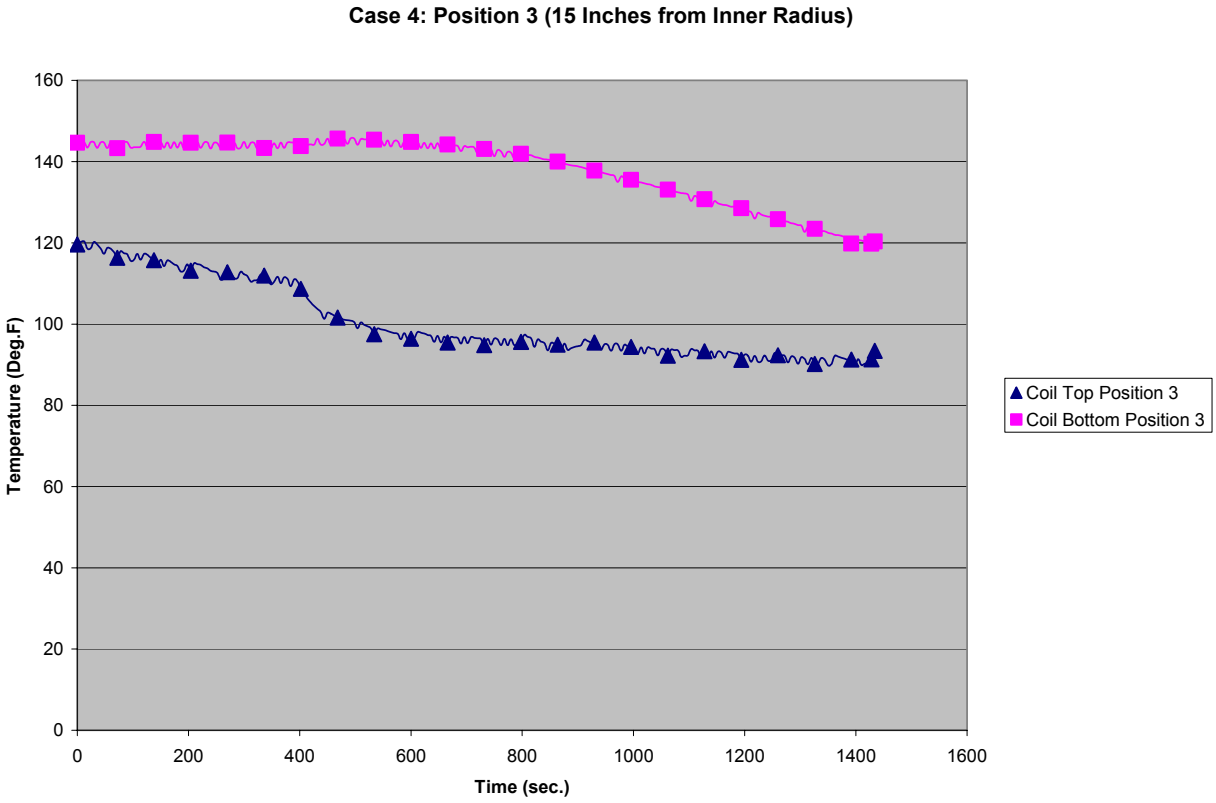

Case 3: Position 4 (21 Inches from Inner Radius)
Temperature (deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 4
Coil Bottom Position 4

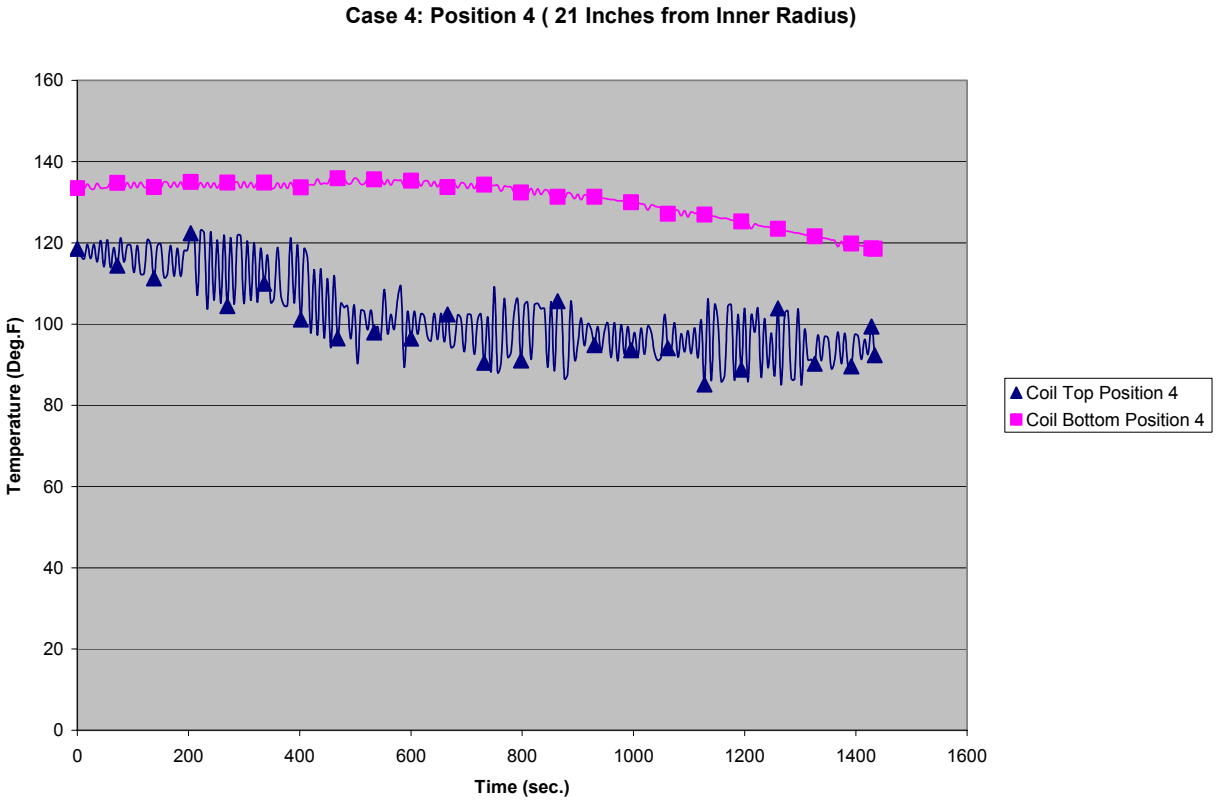
APPENDIX D

Experimental Data for Case 4



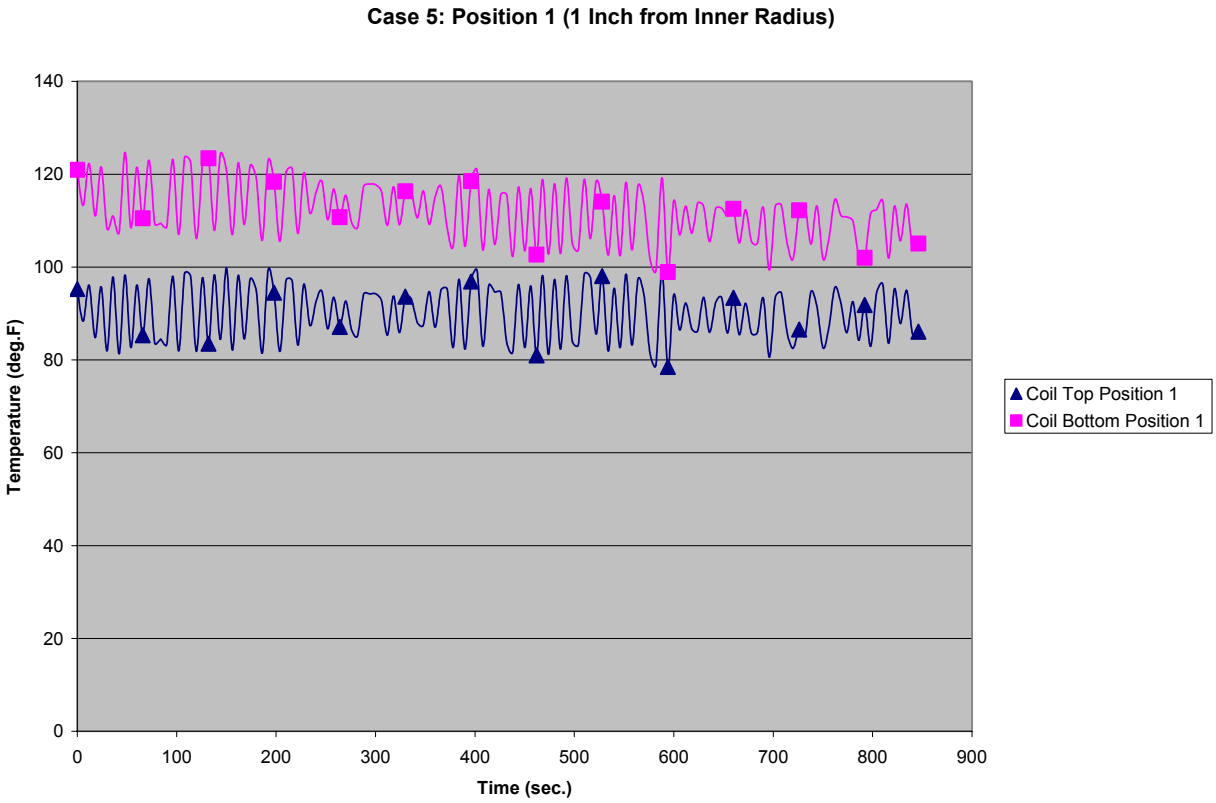

Case 4: Position 2 (9 Inches from Inner Radius)
Temperature (Deg.F)
180
160
140
120
100
80
60
40
20
0
0
200
400
600
800
1000
1200
1400
1600
Time (sec.)
Coil Top Position 2
Coil Bottom Position 2

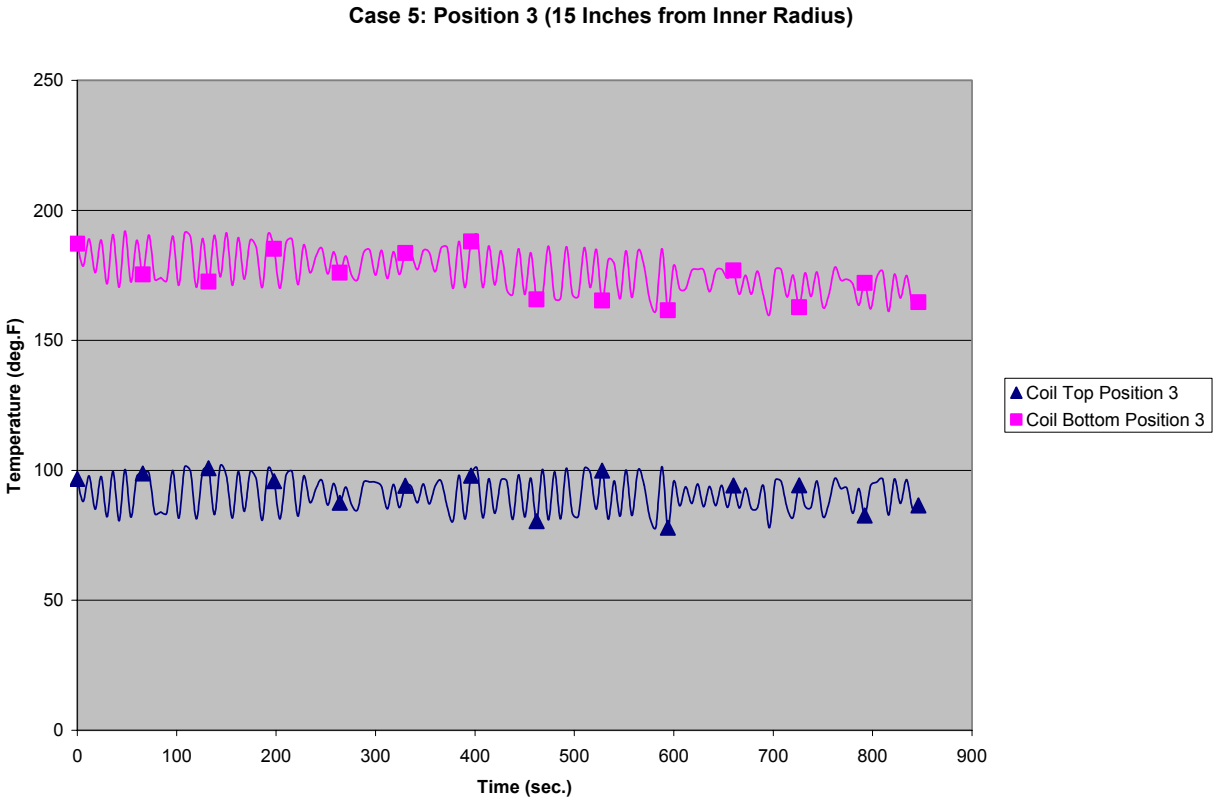

Case 4: Position 3 (15 Inches from Inner Radius)
Temperature (Deg.F)
160
140
120
100
80
60
40
20
0
0
200
400
600
800
1000
1200
1400
1600
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3


Case 4: Position 4 (21 Inches from Inner Radius)
Temperature (Deg.F)
160
140
120
100
80
60
40
20
0
0
200
400
600
800
1000
1200
1400
1600
Time (sec.)
Coil Top Position 4
Coil Bottom Position 4

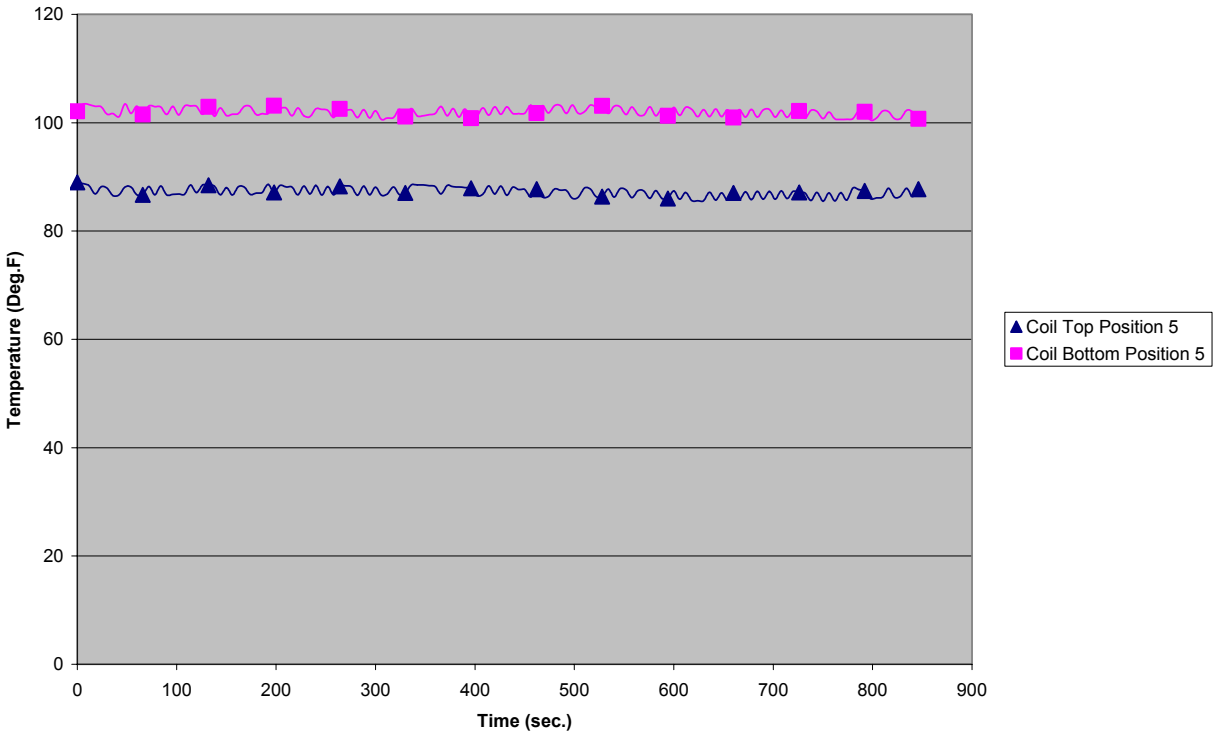
APPENDIX E

Experimental Data for Case 5



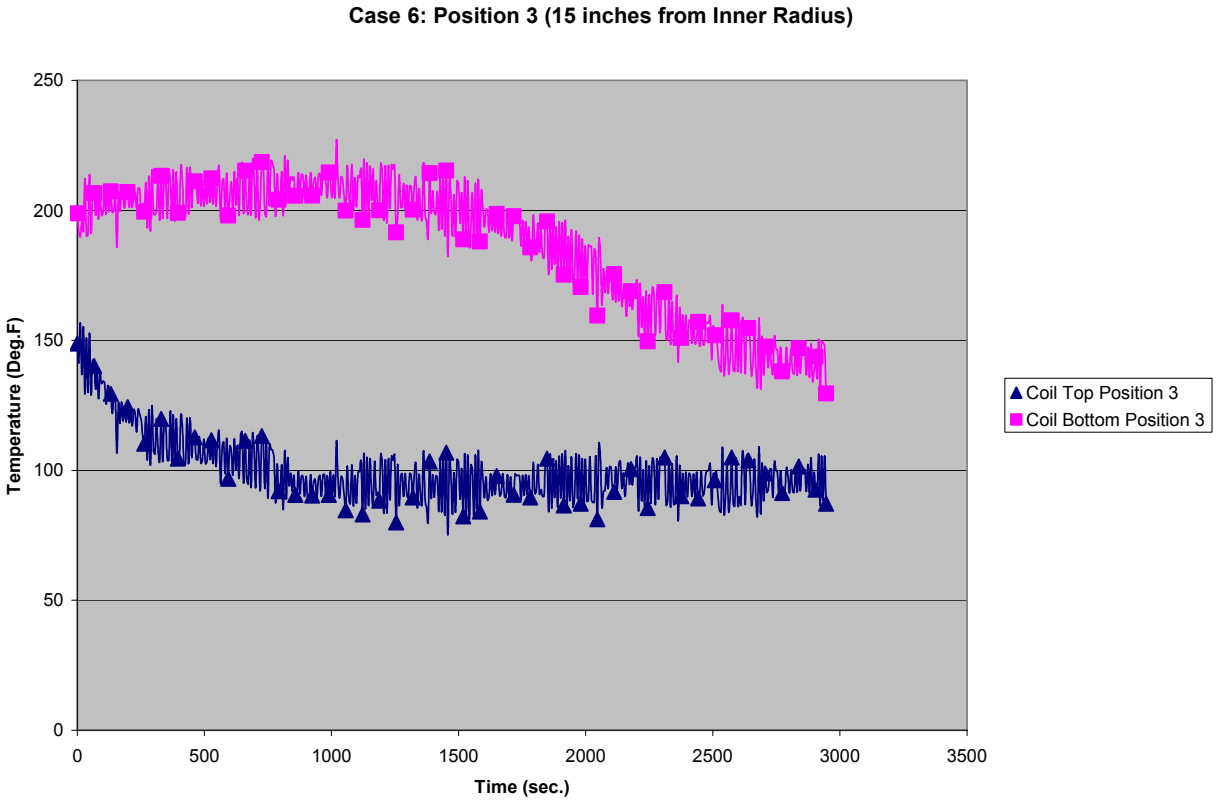

Case 5: Position 3 (15 Inches from Inner Radius)
Temperature (deg.F)
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3

Case 5: Position 5 (27 Inches from Inner Radius)

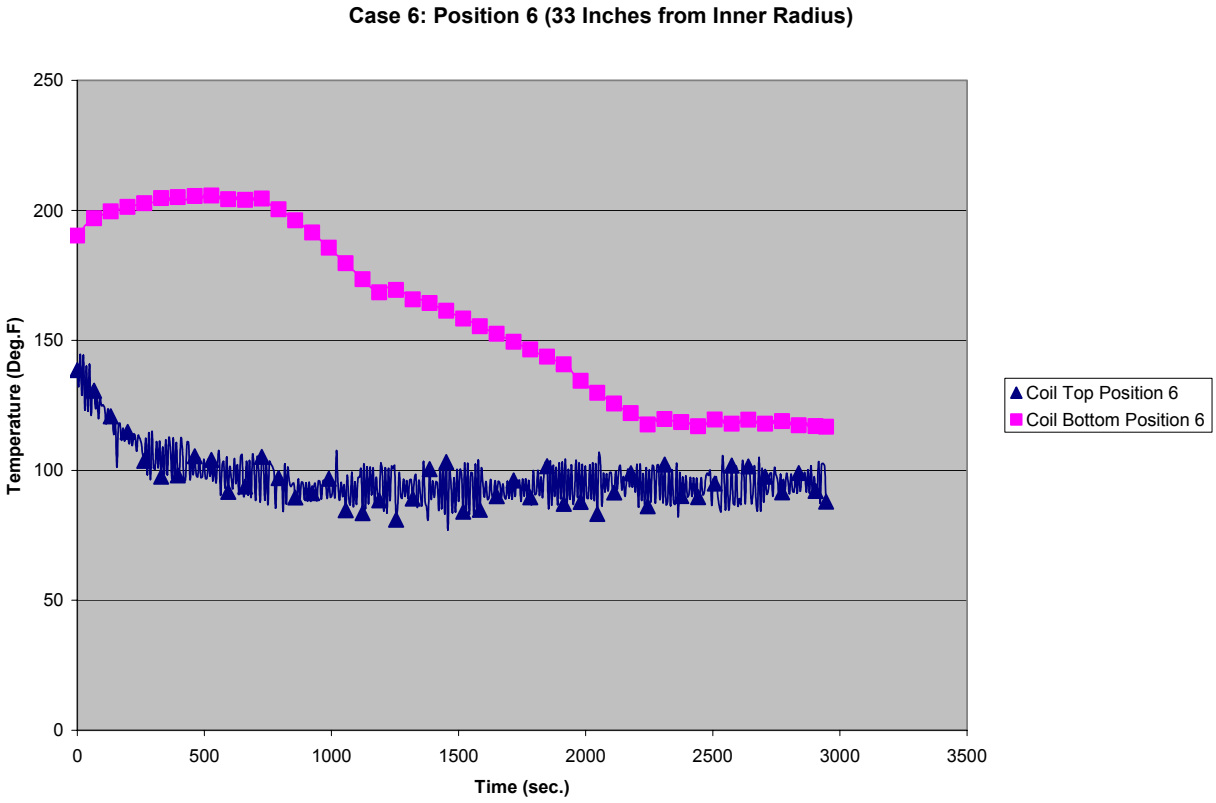


APPENDIX F

Experimental Data for Case 6

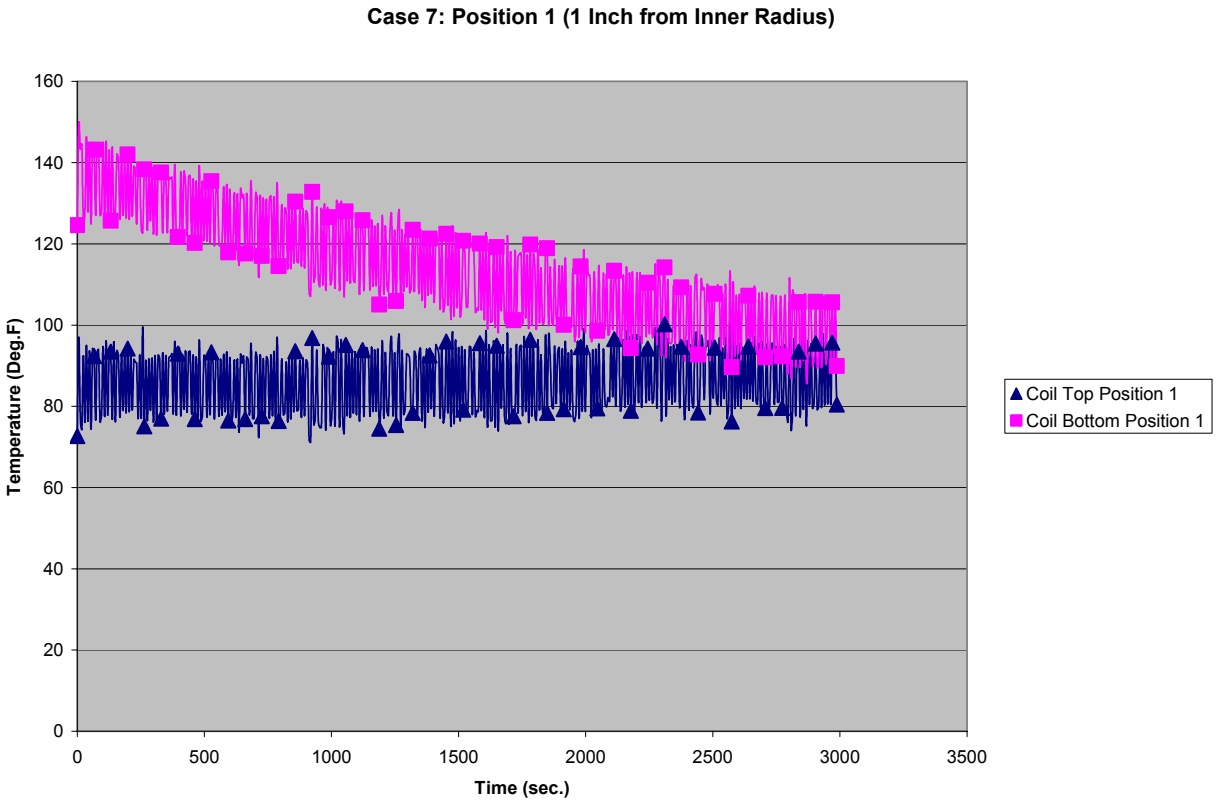

Case 6: Position 3 (15 inches from Inner Radius)
250
200
150
100
50
0
Temperature (Deg.F)
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3

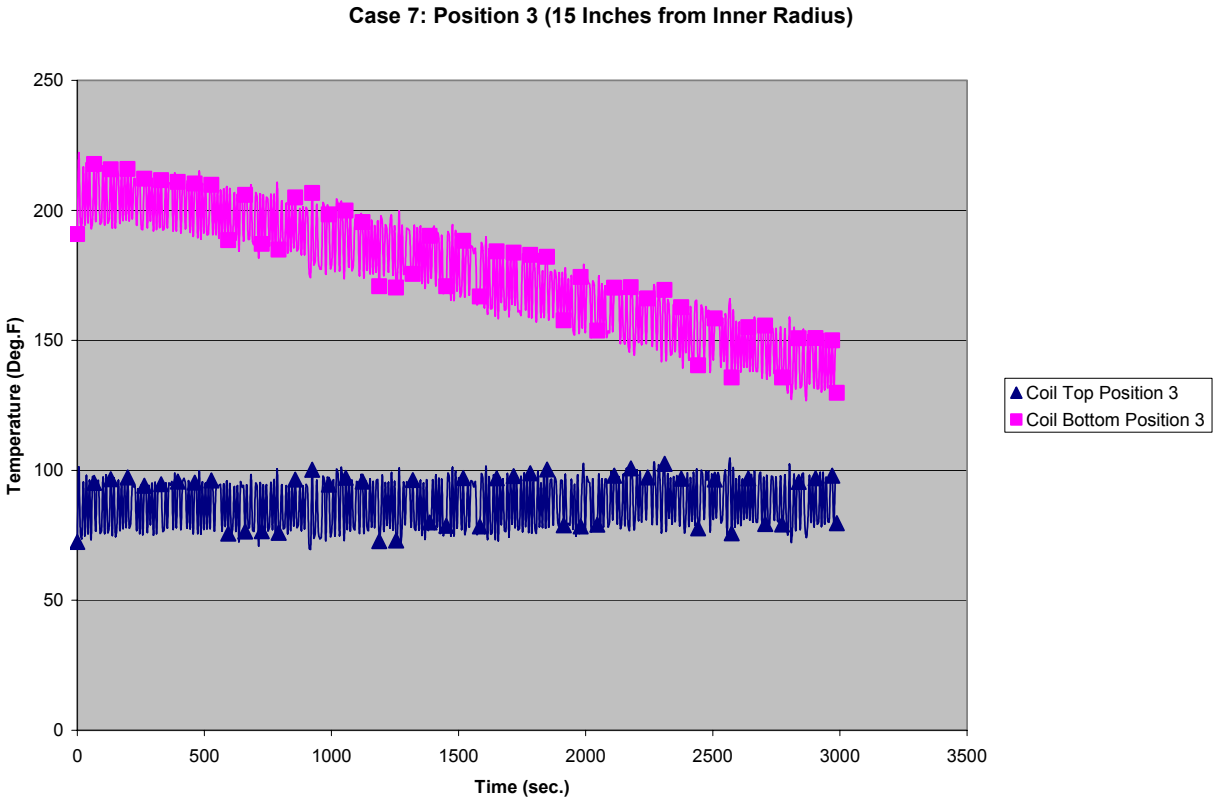

Case 6: Position 4 (21 Inches from Inner Radius)
Temperature (Deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 4
Coil Bottom Position 4

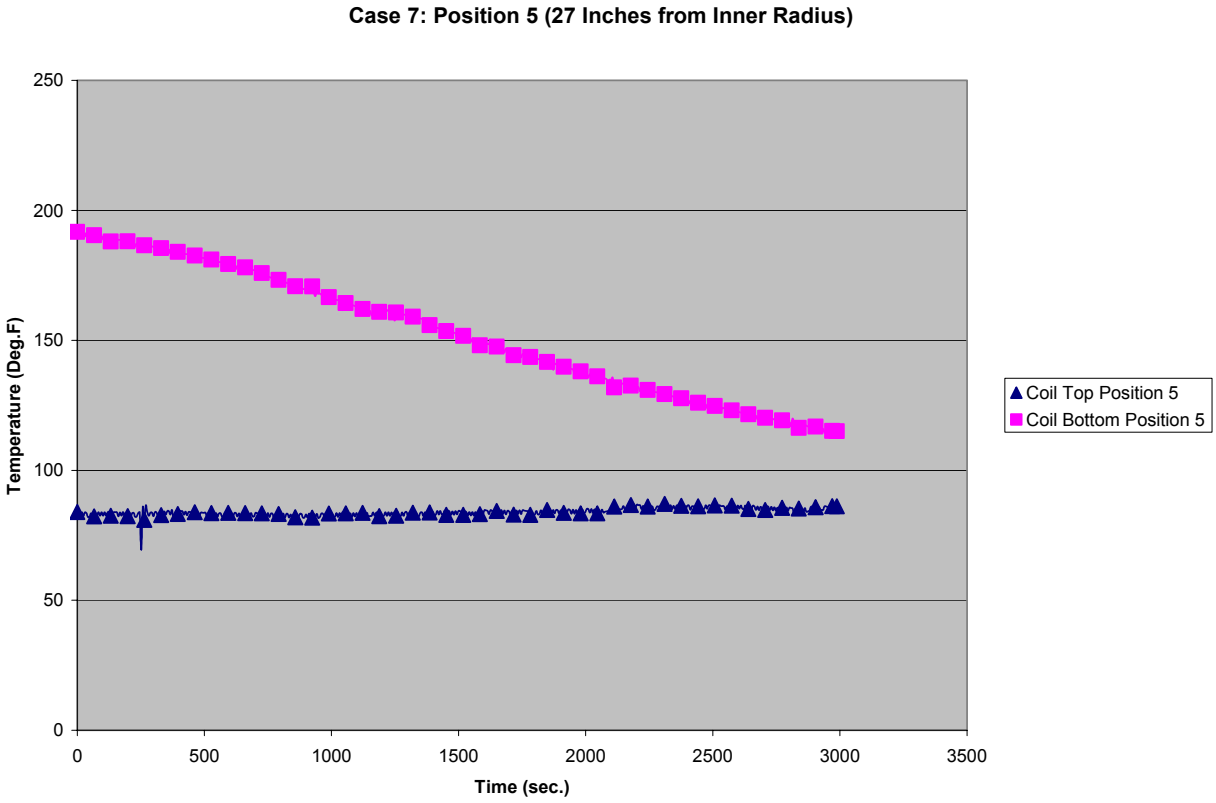

Case 6: Position 6 (33 Inches from Inner Radius)
Temperature (Deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 6
Coil Bottom Position 6

APPENDIX G

Experimental Data for Case 7




Case 7: Position 3 (15 Inches from Inner Radius)
Temperature (Deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 3
Coil Bottom Position 3


Case 7: Position 5 (27 Inches from Inner Radius)
Temperature (Deg.F)
250
200
150
100
50
0
0
500
1000
1500
2000
2500
3000
3500
Time (sec.)
Coil Top Position 5
Coil Bottom Position 5

APPENDIX H

Computer Model for OCA Cooling Analysis

```
% FILE:  oca_cooling_model.m

% Analyzes a full model of 300 radial wraps
% and 1 node per inch of width

% Select output file data
fid_temp = fopen('a:temp.dat','w');

% Determine the room and initial coil temperatures
t_in_F      = input('\nWhat is the room (ambient) temperature in deg.F? \n');
t_init_F    = input('What is the initial bottom (on cooling bed) coil temperature in deg.F?
\n');
t_top_F     = input('What is the initial top (on cooling bed) coil temperature in deg.F? \n');
gauge_inch  = input('What is the gauge of the coil in inches? \n');
width_inch  = input('What is the width of the coil to the nearest inch? \n');

fprintf(' \n')

% U.S. Customary to Metric conversion
t_in_C      = (5/9)*(t_in_F - 32);      %air inlet temp.          (deg.C)
t_init_C    = (5/9)*(t_init_F - 32);    %steel init. bottom temp. (deg.C)
t_top_C     = (5/9)*(t_top_F - 32);     %steel init. top temp.    (deg.C)
gauge_m     = gauge_inch*0.0254;        %thick. of wrap           (m)
width_m     = width_inch*0.0254;        %width of coil            (m)

t_delta     = t_init_C - t_top_C;
t_room      = t_in_C;

% Coil data
wraps       = 300;                      %no. of steel wraps
eqns        = 2*wraps - 1;              %no. of wraps and channels
r_inner     = 0.4318;                   %inner rad. of coil       (m)
spacing_m   = 0.001778;                 %air gap between wraps    (m)

% Finite difference data
nodes       = width_inch;               %no. of nodes
```

```
node_hot      = round(width_inch)/4;      %1st full init. temp. node
dx            = width_m/nodes;            %length of one node          (m)
time_steps    = 498;                      %no. of time steps
dt            = 6.0;                      %length of time step         (sec.)

% Constants for air
rho_a         = 1.00;                     %air density                 (kg/m^3)
c_a           = 1000;                     %air specific heat           (J/kg-deg.C)
u             = 0.75;                     %air velocity                (m/sec.)

% Constants for steel
rho_s         = 7850;           %steel density                         (kg/m^3)
c_s           = 434;            %steel specific heat                   (J/kg-deg.C)
h             = 25;             %heat transfer coeff.                  (W/m^2-deg.C)
h_ex          = 25;             %h. t. coeff. out of coil              (W/m^2-deg.C)

% Create temperature arrays
t             = zeros(eqns,nodes);  %creates a 2-D array of zeros (i by n)
t_F           = zeros(eqns,nodes);
t_nodes       = zeros(1,eqns);      %creates a 2-D array of zeros (1 by i)
t_nodes_old   = zeros(1,eqns);
t_nodes_up    = zeros(1,eqns);
a             = zeros(1,eqns);
b             = zeros(1,eqns);
c             = zeros(1,eqns);
d             = zeros(1,eqns);
daa           = zeros(1,eqns);
dab           = zeros(1,eqns);
ds            = zeros(1,eqns);
dsa           = zeros(1,eqns);
dsb           = zeros(1,eqns);

t299          = zeros(1,nodes);     %creates a 2-D array of zeros (1 by n)

% Call geometry and constant functions
[area,p_in,p_out] = ...
        do_geometry(r_inner,gauge_m,spacing_m,eqns);
[a,b,c,dsa,dsb,ds,daa,dab] = ...
        constants_abcd(rho_s,c_s,dx,dt,h,h_ex,rho_a,c_a,u,area,p_in,p_out,eqns);

        time = 0;
        fprintf('time = %g sec. : \n',time)
        for i = 1:eqns
                for n = 1:nodes
                        if n <= node_hot
                                t(i,n) = t_top_C + t_delta*((n - 1)/(node_hot - 1));
```

```
                else
                        t(i,n) = t_init_C;
                end
                t_F(i,n) = t(i,n)*(9/5) + 32;
        end
        t_nodes_up(i) = t_room;
end

%temp. array for the 299th eqn. (150th steel wrap)
%mid-radius of coil
t299_01    = t_F(299,1);      %temp. of first node
t299_nodes = t_F(299,nodes);  %"      " last "  "
fprintf('%g      %g      %g    %4.2f     %4.2f \n'...
        ,0,time,299,t299_01,t299_nodes);
fprintf(fid_temp,'%g     %g      %g    %4.2f    %4.2f \n'...
        ,0,time,299,t299_01,t299_nodes);

% Time step loop
for m = 1:time_steps
        time(m) = m*dt;
        fprintf('time = %g sec. : \n',time(m))
        % Nodal temperature loop
        for i = 1:eqns
                t_nodes_up(i) = t_room;
        end
        for n = 1:nodes
                t_nodes_old = t(:,n);
                t_nodes = nodecalc_option(t_nodes_old,t_nodes_up,t_room,eqns,...
                        a,b,c,dsa,dsb,ds,daa,dab);
                %t_nodes
                t(:,n) = t_nodes';
                if n < nodes
                        t_nodes_up = t_nodes';
                end
        end

        % Print out the temperature field for this time step
        for i = 1:eqns
                for n = 1:nodes
                        t_F(i,n) = t(i,n)*(9.0/5.0) + 32.0;
                end
        end

        %temp. array for the 299th eqn. (150th steel wrap)
        %mid-radius of coil
        t299_01 = t_F(299,1);         %temp. of first node
```

```
        t299_nodes = t_F(299,nodes);  %"      " tenth "  "
        fprintf('%g      %g     %g    %4.2f     %4.2f \n'...
                ,m,time(m),299,t299_01,t299_nodes);
        fprintf(fid_temp,'%g     %g     %g    %4.2f    %4.2f \n'...
                ,m,time(m),299,t299_01,t299_nodes);

end

fclose(fid_temp);
```

```
%do_geometry

%function to determine the surface area, inner and outer perimeters
%of each steel wrap and air channel

function [area,p_in,p_out] = do_geometry(r_inner,gauge_m,spacing_m,eqns)

for i = 1:eqns
        if i == 1
                %innermost steel wrap
                r_in(i)  = r_inner;
                r_out(i) = r_in(i) + gauge_m;
        elseif rem(i,2) == 0
                %air channels
                r_in(i)  = r_out(i-1);
                r_out(i) = r_in(i) + spacing_m;
        else
                %all other steel wraps
                r_in(i)  = r_out(i-1);
                r_out(i) = r_in(i) + gauge_m;
        end
        area(i)  = pi*(r_out(i)^2 - r_in(i)^2);
        p_in(i)  = 2*pi*r_in(i);
        p_out(i) = 2*pi*r_out(i);
end
```

```
%constants_abcd

% Function to determine the "a", "b', "c" and "d" constants
% for each steel wrap and air channel for the tdma solver
% For 300 wrap boundary condition

function [a,b,c,dsa,dsb,ds,daa,dab] = constants_abcd(rho_s,c_s,dx,dt,h,h_ex,...
        rho_a,c_a,u,area,p_in,p_out,eqns)

for i = 1:eqns
        %steel wrap "sub-constants"
        one_s(i) = rho_s*c_s*area(i)*dx/dt;
        two_s(i) = h*p_in(i)*dx;
        tri_s(i) = h*p_out(i)*dx;
        %air channel "sub-constants"
        one_a(i) = rho_a*c_a*area(i)*dx/dt;
        two_a(i) = rho_a*c_a*area(i)*u;
        if i == 1
                %innermost steel wrap
                a(i)   = one_s(i) + two_s(i)*h_ex/h + tri_s(i);
                b(i)   = tri_s(i);
                c(i)   = 0;
                dsa(i) = one_s(i);
                dsb(i) = two_s(i)*h_ex/h;
                %d(i) = dsa(i)*t_nodes_old(i) + dsb(i)*t_room;
        elseif i == eqns
                %outermost steel wrap
                a(i)   = one_s(i) + two_s(i) + tri_s(i)*h_ex/h;
                b(i)   = 0;
                c(i)   = two_s(i);
                dsa(i) = one_s(i);
                dsb(i) = tri_s(i)*h_ex/h;   %d(i) = dsa(i)*t_nodes_old(i) + dsb(i)*t_room;
                daa(i) = 0.0;
                dab(i) = 0.0;
        elseif rem(i,2) == 0
                %air channels
                a(i)   = one_a(i) + two_a(i) + h*p_out(i-1)*dx + h*p_in(i+1)*dx;
                b(i)   = h*p_in(i+1)*dx;
                c(i)   = h*p_out(i-1)*dx;
                daa(i) = one_a(i);
                dab(i) = two_a(i);   %d(i) = daa(i)*t_nodes_old(i) + dab(i)*t_nodes_up(i);
        else
                %all other steel wraps
                a(i)   = one_s(i) + two_s(i) + tri_s(i);
                b(i)   = tri_s(i);
```

```
            c(i)   = two_s(i);
            ds(i)  = one_s(i);   %d(i) = ds(i)*t_nodes_old(i);
            dsa(i) = 0.0;
            dsb(i) = 0.0;
            daa(i) = 0.0;
            dab(i) = 0.0;
        end
end
```

```
function t_nodes = nodecalc_option(t_nodes_old,t_nodes_up,t_room,eqns,...
        a,b,c,dsa,dsb,ds,daa,dab);

% FILE:  NODECALC_OPTION.M
% Determine a set of nodal temperatures

d(1)    = dsa(1)*t_nodes_old(1) + dsb(1)*t_room(1);
eqm1    = eqns - 1;

for i = 2:eqm1
        % If i =1, NEED TO SET UP t_room PART OF d(i)
        if rem(i,2) == 0   %air channels
                d(i) = daa(i)*t_nodes_old(i) + dab(i)*t_nodes_up(i);
        else            %steel wraps
                d(i) = ds(i)*t_nodes_old(i);
        end
end

% Last wrap
d(eqns) = dsa(eqns)*t_nodes_old(eqns) + dsb(eqns)*t_room;

%use tdma solver to determine t_nodes(i)'s
t_nodes = tdma(a,b,c,d,eqns);
```

```
function x = tdma(a,b,c,d,n_eqns)
%    FILE:  TDMA.M
%    Usage:  x = tdma(a,b,c,d,n_eqns)
%    Subroutine for TDMA solution procedure for tridiagonal, linear equations
%    Typical Eqn. Form:  a(i)*x(i) = b(i)*x(i+1) + c(i)*x(i-1) + d(i)
%    This routine does not change the a,b,c, & d arrays.
%    Solution of the equation set is returned in array x.

%    Solve for the coefficients, p(i) and q(i)

     p(1) = b(1)/a(1);
     q(1) = d(1)/a(1);

     for i = 2:n_eqns
         t = 1./(a(i)-c(i)*p(i-1));
         p(i) = b(i)*t;
         q(i) = (d(i)+c(i)*q(i-1))*t;
     end

%    Solve for x's by back substitution  *

     x(n_eqns) = q(n_eqns);
     for i = 2:n_eqns
         j = n_eqns+1-i;
         x(j) = p(j)*x(j+1) + q(j);
     end
```

```
% FILE:  plot_temp.m

% Plot temps. for top and bottom of coil vs. time

% Get input from a file.
path(path,'a:temp.dat');          % Specify to path to data file if needed.
load temp.dat;                    % Get a matrix from file a:temp.dat
n_pts = length(temp);             % Number of data points in each vector

% Set up vectors for the input data
t       = temp(1:n_pts,2);     % Data from  2nd col. of input file (time)
n1      = temp(1:n_pts,4);     % "       "  4th "              " (first   node temps.)
n10     = temp(1:n_pts,5);     % "       "  5th "              " (tenth   "        ")

% Output the Data
plot(t,n1,t,n10)
axis([0,3000,75,250])
grid                                                % Turn on grid lines
xlabel('t (time)')                                  % x-axis label
ylabel('Temp. (deg.F)')                             % y-axis label
title('Temps. vs. Time for Top and Bottom of Coil') % Title of plots
```

APPENDIX I

Computer Model for the Study of Sitting on Floor Model

```
%tfldwall_axialx

%Program to calulate the temperature distribution in a channel and wall
%under transient conditions with heat transfer from a solid wall
%releasing its stored heat to the fluid
%Axial conduction version

%Examines temperature distribution along width for coil sitting on floor

%setup the input values
nodes       = 25;                       %number of nodes across length Xmax
max_steps   = 1000;                     %maximum number of time steps for dt
time        = zeros(1,max_steps);
n_out       = 10;                       %number of values in multiline output array
n_lines     = ceil(nodes/n_out);        %number of lines of values in output
n_profiles  = 6;                        %number of profile plots
t_profile   = zeros(n_profiles,nodes);
profile_step = [0,ceil(0.2*max_steps),ceil(0.4*max_steps),...
                ceil(0.6*max_steps),ceil(0.8*max_steps),max_steps];

x           = zeros(1,nodes);

rho         = 1.00;         %density of fluid,              (kg/m^3)
c           = 1000;         %specific heat of fluid,        (J/kg-deg.C)
rho_w       = 7850;         %density of solid,              (kg/m^3)
c_w         = 434;          %specific heat of solid,        (J/kg-deg.C)
k_w         = 60.50;        %th. cond. of solid,            (W/m-deg.C)
gauge       = .000610;      %Metal gauge thickness,         (m)
gap         = 0.001778;     %Gap spacing,                   (m)
D           = 2.0*gap;      %Hydraulic diameter,            (m)
D_t         = D + gauge;    %outside diameter of tube,      (m)
u           = 0.01;         %velocity of fluid,             (m/sec.)
h           = 25;           %h. t. coeff. of fluid,         (W/m^2-deg.C)
h_exL       = 100;          %Left h. t. coeff. of external, (W/m^2-deg.C)
h_exR       = 10;           %Right h. t. coeff. of external,(W/m^2-deg.C)
END_HTL     = 1;            %HT from left end, Yes = 1
```

```
END_HTR     = 1;            %HT from right end, Yes = 1

t_init      = 100.0;        %initial temperature of solid,      (deg.C)
tin         = 0.0;          %inlet temperature of fluid,        (deg.C)
Xmax        = 1.0;          %length of solid,                   (m)
dt          = 1.0;          %time step,                         (sec.)

%determine some constants for the problem
dx    = Xmax/nodes;         %change in x
x(1) = dx/2.0;

for j = 2:nodes
        x(j) = x(j-1) + dx;
end

pi      = 4*atan(1.0);              %equation to determine pi
P       = pi*D;                     %perimeter of inner wall,       (m)
A       = pi*D*D/4;                 %flow area,                     (m^2)
A_w     = pi*(D_t^2 - D^2)/4;       %area of wall normal to flow,   (m^2)

%print out header values
fprintf('\n  Transient Flow in a Channel, Fluid-Wall Interaction \n \n')
fprintf('  number of nodes              = %4g \n',nodes)
fprintf('  Number of time steps         = %4g \n',max_steps)
fprintf('  max. length                  = %6.3f (m) \n',Xmax)
fprintf('  delta x                      = %6.3f (m) \n',dx)
fprintf('  delta t                      = %6.3f (sec) \n \n',dt)
fprintf('  Gauge thickness              = %6.3f (mm) \n',gauge*1000.)
fprintf('  Gap spacing                  = %6.3f (mm) \n',gap*1000.)
fprintf('  hydraulic diameter           = %6.3f (mm) \n',D*1000.)
fprintf('  tube O.D.                    = %6.3f (mm) \n \n',D_t*1000.)
fprintf('  fluid density                = %6.3f (kg/cu.m) \n',rho)
fprintf('  fluid sp. heat               = %5.0f (J/kg-deg.C) \n',c)
fprintf('  wall density                 = %5.0f (kg/cu.m) \n',rho_w)
fprintf('  wall sp. heat                = %5.0f (J/kg-deg.C) \n',c_w)
fprintf('  wall th. cond.               = %6.3f (W/m-deg.C) \n',k_w)
fprintf('  velocity                     = %6.3f (m/sec) \n',u)
fprintf('  h. t. coefficient            = %6.1f (W/sq.m-deg.C) \n',h)
fprintf('  Ext. H.T. switch, Left       = %2.0f (On =1, Off =0)\n',END_HTL)
fprintf('  ext. h. t. coeff., Left  ht  = %2.0f (On =1, Off =0)\n',END_HTR)
fprintf('  ext. h. t. coeff., Right     = %6.1f (W/sq.m-deg.C) \n',h_exR)
fprintf('  initial wall temp.           = %6.2f  (deg.C) \n',t_init)
fprintf('  initial fluid temp.          = %6.2f  (deg.C) \n',t_init)
fprintf('  inlet temp.                  = %6.2f  (deg.C) \n \n',tin)

fprintf('press any key to continue \n \n')
```

```
pause

c_P             = rho*c*A*u;                %density*sp. heat*area*velocity, (fluid)
d_P             = rho*c*A*dx/dt;            %density*sp.heat*A*(dx/dt),  (fluid)
e_P             = h*P*dx;                   %h. t. coeff.*perimeter*dx   (fluid)
a_P             = d_P + c_P + e_P;
dd_P            = rho_w*c_w*A_w*dx/dt;      %density*sp. heat*area*(dx/dt) (wall)
kAdiv_dx        = k_w*A_w/dx;

if (k_w < 0.0001)|(h_exL < 0.0001)
        R_eqL = 1.0e10;
else
        R_eqL = (1.0/h_exL + dx/2/k_w)/A_w;
end
if (k_w < 0.0001)|(h_exR < 0.0001)
        R_eqR = 1.0e10;
else
        R_eqR = (1.0/h_exR + dx/2/k_w)/A_w;
end

a       = a_P*ones(1,nodes); %Set sizes & some values of coeff. arrays
b       = zeros(1,nodes);
c       = c_P*ones(1,nodes);
d       = zeros(1,nodes);
e       = e_P*ones(1,nodes);

bb      = kAdiv_dx*ones(1,nodes);
cc      = kAdiv_dx*ones(1,nodes);
dd      = zeros(1,nodes);
ee      = e_P*ones(1,nodes);

for i = 1:nodes
        if i == 1
                aa(i) = dd_P + bb(i) + ee(i);
                if END_HTL == 1
                        aa(i) = aa(i) + 1/R_eqL;
                end
        elseif i == nodes
                aa(i) = dd_P + cc(i) + ee(i);
                if END_HTR == 1
                        aa(i) = aa(i) + 1/R_eqR;
                end
        else
                aa(i) = dd_P + bb(i) + cc(i) + ee(i);
        end
end
```

```
fprintf('  T(1)   T(2)   T(3)   T(4)   T(5)   T(6)   T(7)   T(8)   T(9)   T(10)\n')

%main calculation loop
for istep = 0:max_steps;
        if istep == 0
                time_start = 0;
                for i = 1:nodes
                        tp(i)      = t_init;
                        tp_wall(i) = t_init;
                end
                tp_F = tp*(9.0/5.0)+32.0;
                tp_wall_F = tp_wall*(9.0/5.0)+32.0;
                t_profile(1,:) = tp_wall_F;
                fprintf('  time = %g (sec), step = %g, nodes = %g, fluid/wall \n',...
                        time_start,istep,nodes)
                for line = 1:n_lines
                        t_out = multiline(tp_F,nodes,n_out,line);
                        fprintf('  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f
                        %6.2f\n'...
                                ,t_out(1:n_out))
                end %for
                for line = 1:n_lines
                        t_out = multiline(tp_wall_F,nodes,n_out,line);
                        fprintf('  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %6.2f
                        %6.2f\n'...
                                ,t_out(1:n_out))
                end %for
        else
                time(istep) = istep*dt;
                for i = 1:nodes
                        t_old(i)     = tp(i);
                        twall_old(i) = tp_wall(i);
                        if i == 1
                                c(1) = 0.0;
                                cc(1) = 0.0;
                                d(1) = d_P*t_old(1) + c_P*tin;
                                dd(1) = dd_P*twall_old(1);
                                if END_HTL == 1
                                        dd(1) = dd(1) + (1/R_eqL)*tin;
                                end
                        elseif i == nodes
                                b(i) = 0.0;
                                bb(i) = 0.0;
                                d(i) = d_P*t_old(i);
                                dd(i) = dd_P*twall_old(i);
```

```
                        if END_HTR == 1
                                dd(i) = dd(i) + (1/R_eqR)*tin;
                        end
                else
                        d(i) = d_P*t_old(i);
                        dd(i) = dd_P*twall_old(i);
                end
                xdata(istep,i) = time(istep);
        end
        [tp,tp_wall] = dtdma(a,b,c,d,e,aa,bb,cc,dd,ee,nodes);
        tp_F = tp*(9.0/5.0) + 32.0;
        tp_wall_F = tp_wall*(9.0/5.0)+32.0;
        for jplot = 2:5
                if istep == profile_step(jplot)
                t_profile(jplot,:) = tp_wall_F;
                end
        end
        fprintf('  time = %g (sec), step = %g, nodes = %g, fluid/wall \n',...
                time(istep),istep,nodes)
        for line = 1:n_lines
                t_out = multiline(tp_F,nodes,n_out,line);
                fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                %6.2f\n'...
                        ,t_out(1:n_out))
        end %for
        for line = 1:n_lines
                t_out = multiline(tp_wall_F,nodes,n_out,line);
                fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                %6.2f\n'...
                        ,t_out(1:n_out))
        end %for
        ydatatp(istep,:) = tp;
        ydatatp_wall(istep,:) = tp_wall;
    end
end

t_profile(6,:) = tp_wall_F;
plot(x,t_profile(1,:),x,t_profile(2,:),x,t_profile(3,:),...
    x,t_profile(4,:),x,t_profile(5,:),x,t_profile(6,:))
grid
xlabel('x(m)')
ylabel('Wall Temp.(F)')
axis([0 1.0 0 250])
%subplot(1,2,1),plot(xdata,ydatatp,'b-'),xlabel('time (s)')...
%   ,ylabel('fluid temp'),title('tp'),grid;
%subplot(1,2,2),plot(xdata,ydatatp_wall,'b-'),xlabel('time (s)')...
```

```
%   ,ylabel('wall temp'),title('tp_w_a_l_l'),grid;
```

```
function y = multiline(x,n_x,n_y,line)

%       FILE:  multiline.m
%       Usage:  y = multiline(x,n_x,n_y,line)
%       Function to output part of a long array in a smaller one.
%       x, n_x -- Large array and its size
%       y, n_y -- Small array and its size
%       line  -- location of values in small array
%       Last set of array y are -1 when beyond n_x

first   = (line - 1)*n_y + 1;
last    = first + n_y - 1;

if last < n_x
        y = x(first:last);
else
        for i = 1:n_y
                j = (line - 1)*n_y + i;
                if j <= n_x
                        y(i) = x(j);
                else
                        y(i) = -1.0;
                end %if
        end %for
end %if
```

```
function [f,g] = dtdma(a,b,c,d,e,aa,bb,cc,dd,ee,n_eqns)

%       FILE:  dtdma.m
%       Usage:  [f,g] = dtdma(a,b,c,d,e,aa,bb,cc,dd,ee,n_eqns)
%       Subroutine for TDMA solution procedure for a double tridiagonal, linear equations
%       Typical Eqn. Form:    a(i)*f(i) = b(i)*f(i+1) + c(i)*f(i-1) + d(i) + e(i)*g(i)
%                             aa(i)*g(i) = bb(i)*g(i+1) + cc(i)*g(i-1) + dd(i) + ee(i)*f(i)
%       This routine does not change the a,b,c,d,e & aa,bb,cc,dd,ee arrays.
%       Solution of the equation set is returned in arrays f & g.

%       Solve for the coefficients when i = 1

        delta(1)        = a(1)*aa(1) - e(1)*ee(1);
        p(1)            = aa(1)*b(1)/delta(1);
        pp(1)           = a(1)*bb(1)/delta(1);
        q(1)            = bb(1)*e(1)/delta(1);
        qq(1)           = b(1)*ee(1)/delta(1);
        r(1)            = (aa(1)*d(1) + dd(1)*e(1))/delta(1);
        rr(1)           = (a(1)*dd(1) + d(1)*ee(1))/delta(1);

        for i = 2:n_eqns
                delta(i) = (a(i) - c(i)*p(i-1)) * (aa(i) - cc(i)*pp(i-1)) ...
                        -(e(i) + c(i)*q(i-1)) * (ee(i) + cc(i)*qq(i-1));
                p(i)  = (aa(i) - cc(i)*pp(i-1))*b(i)/delta(i);
                pp(i) = (a(i) - c(i)*p(i-1))*bb(i)/delta(i);
                q(i)  = (e(i) + c(i)*q(i-1))*bb(i)/delta(i);
                qq(i) = (ee(i) + cc(i)*qq(i-1))*b(i)/delta(i);
           r(i)  = ((aa(i) - cc(i)*pp(i-1)) * (d(i) + c(i)*r(i-1)) + ...
                        (e(i) + c(i)*q(i-1)) * (dd(i) + cc(i)*rr(i-1)))/delta(i);
           rr(i) = ((a(i) - c(i)*p(i-1)) * (dd(i) + cc(i)*rr(i-1)) + ...
                        (ee(i) + cc(i)*qq(i-1)) * (d(i) + c(i)*r(i-1)))/delta(i);
        end

%   Solve for x's by back substitution  *

        f(n_eqns) = r(n_eqns);
        g(n_eqns) = rr(n_eqns);

        for i = 2:n_eqns
                j = n_eqns+1-i;
                f(j) = p(j)*f(j+1) + q(j)*g(j+1) + r(j);
                g(j) = pp(j)*g(j+1) + qq(j)*f(j+1) + rr(j);
        end
```

APPENDIX J

Computer Model for the Study of the Effect of Conduction

Note: The functions dtdma.m and multiline.m from Appendix I are used for this program also.

```
%tfldwall_axial

%Program to calulate the temperature distribution in a channel and wall
%under transient conditions with heat transfer from a solid wall
%releasing its stored heat to the fluid
%Axial conduction version

%setup the input values
nodes       = 18;                       %number of nodes across length Xmax
max_steps   = 52;                       %maximum number of time steps for dt
time        = zeros(1,max_steps);
n_out       = 10;                       %number of values in multiline output array
n_lines     = ceil(nodes/n_out);        %number of lines of values in output
n_profiles  = 6;                        %number of profile plots
t_profile   = zeros(n_profiles,nodes);

profile_step = [0,ceil(0.2*max_steps),ceil(0.4*max_steps),...
        ceil(0.6*max_steps),ceil(0.8*max_steps),max_steps];

x = zeros(1,nodes);

rho         = 1.00;         %density of fluid,          (kg/m^3)
c           = 1000;         %specific heat of fluid,    (J/kg-deg.C)
rho_w       = 7850;         %density of solid,          (kg/m^3)
c_w         = 434;          %specific heat of solid,    (J/kg-deg.C)
k_w         = 60.50;        %th. cond. of solid,        (W/m-deg.C)
gauge       = .000610;      %Metal gauge thickness,     (m)
gap         = 0.001778;     %Gap spacing,               (m)
D           = 2.0*gap;      %Hydraulic diameter,        (m)
D_t         = D + gauge;    %outside diameter of tube,  (m)
u           = 1.00;         %velocity of fluid,         (m/sec.)
h           = 25;           %h. t. coeff. of fluid,     (W/m^2-deg.C)
h_ex        = 100;          %h. t. coeff. of external,  (W/m^2-deg.C)
END_HTL     = 1;            %HT from left end, Yes = 1
```

```
END_HTR     = 1;              %HT from right end, Yes = 1

t_init   = 100.0;       %initial temperature of solid, (deg.C)
tin      = 0.0;         %inlet temperature of fluid,   (deg.C)
Xmax  = 1.0;            %length of solid,              (m)
dt       = 5.0;         %time step,                    (sec.)

%determine some constants for the problem
dx       = Xmax/nodes;          %change in x
x(1)     = dx/2.0;

for j = 2:nodes
        x(j) = x(j-1) + dx;
end

pi       = 4*atan(1.0);                 %equation to determine pi
P        = pi*D;                        %perimeter of inner wall,        (m)
A        = pi*D*D/4;                    %flow area,                      (m^2)
A_w    = pi*(D_t^2 - D^2)/4;            %area of wall normal to flow,    (m^2)

%print out header values
fprintf('\n  Transient Flow in a Channel, Fluid-Wall Interaction \n \n')
fprintf('  number of nodes              = %4g \n',nodes)
fprintf('  Number of time steps         = %4g \n',max_steps)
fprintf('  max. length                  = %6.3f (m) \n',Xmax)
fprintf('  delta x                      = %6.3f (m) \n',dx)
fprintf('  delta t                      = %6.3f (sec) \n \n',dt)
fprintf('  Gauge thickness              = %6.3f (mm) \n',gauge*1000.)
fprintf('  Gap spacing                  = %6.3f (mm) \n',gap*1000.)
fprintf('  hydraulic diameter           = %6.3f (mm) \n',D*1000.)
fprintf('  tube O.D.                    = %6.3f (mm) \n \n',D_t*1000.)
fprintf('  fluid density                = %6.3f (kg/cu.m) \n',rho)
fprintf('  fluid sp. heat               = %5.0f (J/kg-deg.C) \n',c)
fprintf('  wall density                 = %5.0f (kg/cu.m) \n',rho_w)
fprintf('  wall sp. heat                = %5.0f (J/kg-deg.C) \n',c_w)
fprintf('  wall th. cond.               = %6.3f (W/m-deg.C) \n',k_w)
fprintf('  velocity                     = %6.3f (m/sec) \n',u)
fprintf('  h. t. coefficient            = %6.1f (W/sq.m-deg.C) \n',h)
fprintf('  ext. h. t. coeff.            = %6.1f (W/sq.m-deg.C) \n',h_ex)
fprintf('  Ext. H.T. switch, Left       = %2.0f (On =1, Off =0)\n',END_HTL)
fprintf('  Ext. H.T. switch, Right      = %2.0f (On =1, Off =0)\n',END_HTR)
fprintf('  initial wall temp.           = %6.2f  (deg.C) \n',t_init)
fprintf('  initial fluid temp.          = %6.2f  (deg.C) \n',t_init)
fprintf('  inlet temp.                  = %6.2f  (deg.C) \n \n',tin)

fprintf('press any key to continue \n \n')
```

```
pause

c_P             = rho*c*A*u;                %density*sp. heat*area*velocity, (fluid)
d_P             = rho*c*A*dx/dt;            %density*sp.heat*A*(dx/dt),  (fluid)
e_P             = h*P*dx;                   %h. t. coeff.*perimeter*dx   (fluid)
a_P             = d_P + c_P + e_P;
dd_P            = rho_w*c_w*A_w*dx/dt;      %density*sp. heat*area*(dx/dt) (wall)
kAdiv_dx        = k_w*A_w/dx;

if (k_w < 0.0001)|(h_ex < 0.0001)
        R_eq = 1.0e10;
else
        R_eq = (1.0/h_ex + dx/2/k_w)/A_w;
end

a       = a_P*ones(1,nodes); %Set sizes & some values of coeff. arrays
b       = zeros(1,nodes);
c       = c_P*ones(1,nodes);
d       = zeros(1,nodes);
e       = e_P*ones(1,nodes);

bb      = kAdiv_dx*ones(1,nodes);
cc      = kAdiv_dx*ones(1,nodes);
dd      = zeros(1,nodes);
ee      = e_P*ones(1,nodes);

for i = 1:nodes
        if i == 1
                aa(i) = dd_P + bb(i) + ee(i);
                if END_HTL == 1
                        aa(i) = aa(i) + 1/R_eq;
                end
        elseif i == nodes
                aa(i) = dd_P + cc(i) + ee(i);
                if END_HTR == 1
                        aa(i) = aa(i) + 1/R_eq;
                end
        else
                aa(i) = dd_P + bb(i) + cc(i) + ee(i);
        end
end

fprintf('  T(1)   T(2)   T(3)   T(4)   T(5)   T(6)   T(7)   T(8)   T(9)   T(10)\n')

%main calculation loop
for istep = 0:max_steps;
```

```
if istep == 0
        time_start = 0;
        for i = 1:nodes
                tp(i)      = t_init;
                tp_wall(i) = t_init;
        end
        tp_F = tp*(9.0/5.0)+32.0;
        tp_wall_F = tp_wall*(9.0/5.0)+32.0;
        t_profile(1,:) = tp_wall_F;
        fprintf('  time = %g (sec), step = %g, nodes = %g, fluid/wall \n',...
                time_start,istep,nodes)
        for line = 1:n_lines
                t_out = multiline(tp_F,nodes,n_out,line);
                fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                %6.2f\n'...
                        ,t_out(1:n_out))
        end %for
        for line = 1:n_lines
                t_out = multiline(tp_wall_F,nodes,n_out,line);
                fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                %6.2f\n'...
                        ,t_out(1:n_out))
        end %for
else
        time(istep) = istep*dt;
        for i = 1:nodes
                t_old(i)     = tp(i);
                twall_old(i) = tp_wall(i);
                if i == 1
                        c(1) = 0.0;
                        cc(1) = 0.0;
                        d(1) = d_P*t_old(1) + c_P*tin;
                        dd(1) = dd_P*twall_old(1);
                        if END_HTL == 1
                        dd(1) = dd(1) + (1/R_eq)*tin;
                        end
                elseif i == nodes
                        b(i) = 0.0;
                        bb(i) = 0.0;
                        d(i) = d_P*t_old(i);
                        dd(i) = dd_P*twall_old(i);
                        if END_HTR == 1
                                dd(i) = dd(i) + (1/R_eq)*tin;
                        end
                else
                        d(i) = d_P*t_old(i);
```

```
                        dd(i) = dd_P*twall_old(i);
                    end
                    xdata(istep,i) = time(istep);
                end
                [tp,tp_wall] = dtdma(a,b,c,d,e,aa,bb,cc,dd,ee,nodes);
                tp_F = tp*(9.0/5.0) + 32.0;
                tp_wall_F = tp_wall*(9.0/5.0)+32.0;
                for jplot = 2:5
                    if istep == profile_step(jplot)
                        t_profile(jplot,:) = tp_wall_F;
                    end
                end
                fprintf('  time = %g (sec), step = %g, nodes = %g, fluid/wall \n',...
                    time(istep),istep,nodes)
                for line = 1:n_lines
                    t_out = multiline(tp_F,nodes,n_out,line);
                    fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                    %6.2f\n'...
                        ,t_out(1:n_out))
                end %for
                for line = 1:n_lines
                    t_out = multiline(tp_wall_F,nodes,n_out,line);
                    fprintf(' %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f
                    %6.2f\n'...
                        ,t_out(1:n_out))
                end %for
                ydatatp(istep,:) = tp;
                ydatatp_wall(istep,:) = tp_wall;
        end
end
t_profile(6,:) = tp_wall_F;

plot(x,t_profile(1,:),x,t_profile(2,:),x,t_profile(3,:),...
        x,t_profile(4,:),x,t_profile(5,:),x,t_profile(6,:))

grid
xlabel('x(m)')
ylabel('Wall Temp.(F)')
axis([0 1.0 0 250])

%subplot(1,2,1),plot(xdata,ydatatp,'b-'),xlabel('time (s)')...
%   ,ylabel('fluid temp'),title('tp'),grid;
%subplot(1,2,2),plot(xdata,ydatatp_wall,'b-'),xlabel('time (s)')...
%   ,ylabel('wall temp'),title('tp_w_a_l_l'),grid;
```

BIBLIOGRAPHY

BIBLIOGRAPHY

1. "Finite Element analysis of hot rolled coil cooling". S. J. Park, B. H. Hong and K. H. Oh, ISIJ International, Vol. 38, No. 11, p. 1262-1269 (1998).

2. "A two-equation analysis of convection heat transfer in porous media", H. Y. Zhang and X. Y. Huang, Transport in Porous Media, Vol. 44, No. 2, p. 305-324 (Aug 2001).

3. Patankar, Suhas V., Numerical Heat Transfer and Fluid Flow (New York: McGraw-Hill, 1980), pp. 25-100.

4. Patankar, Op. Cit., p. 96-100.

5. Patankar, Op. Cit., p. 52-54.

6. White, Frank M., Fluid Mechanics (Fourth Edition; New York: McGraw-Hill, 1999), pp. 10.

7. Incropera, Frank P., Witt, David P., Introduction to Heat Transfer (Third Edition; New York: John Wiley & Sons, 1996), pp. 8.

8. Incropera, Frank P., Witt, David P., Op. Cit., p. 757.

9. Incropera, Frank P., Witt, David P., Op. Cit., p. 746.

10. Private correspondence with Dr. Richard S. Dougall, University of Pittsburgh.